#19 Sup Spec
NE

SUBSTITUTE SPECIFICATION

OF

U.S. PATENT APPLICATION 09/560,109

ATTORNEY DOCKET NO. M-5628 US

DETECTOR FOR AN ETHERNET RECEIVER

Peter J. Sallaway
Sreen Raghavan

BACKGROUND

1. FIELD OF THE INVENTION

[0001] This invention relates to digital communication systems and, more particularly, to a detection system and method for an Ethernet receiver.

2. DISCUSSION OF RELATED ART

[0002] The dramatic increase in desktop computing power driven by Intranet-based operations and the increased demand for time-sensitive delivery between users has spurred development of high-speed Ethernet LANs. 100BASE-TX Ethernet, using existing category-5 copper wire, and the newly developed 1000BASE-T Ethernet for gigabit/s transfer of data over category-5 copper wire require new techniques in high speed symbol processing. Gigabit per second transfer can be accomplished utilizing four twisted pairs and a 125 megasymbol/s transfer rate on each pair where each symbol represents two bits.

[0003] Physically, data is transferred using a set of voltages where each voltage represents one or more bits of data. Each voltage in the set is referred to as a symbol and the whole set of voltages is referred to as a symbol alphabet. In gigabit transfer, for example, data is usually sent with a set of five voltage levels (PAM-5), each symbol representing two (2) bits.

[0004] One system of transferring data at high rates is Non Return to Zero (NRZ) signaling. In NRZ, the symbol alphabet

{A} is {-1, +1}. A logical "1" is transmitted as a positive voltage while a logical "0" is transmitted as a negative voltage. At 125 M symbols/s, the symbol rate required for gigabit transfer over four category-5 wires, the pulse width of each symbol is 8 ns.

[0005] Another example of a modulation method for high speed symbol transfer is MLT3 and involves a three level system. (*See* American National Standard Information System, *Fibre Distributed Data Interface (FDDI) -Part: Token Ring Twisted Pair Physical Layer Medium Dependent (TP-PMD)*, ANSI X3.263:199X). The symbol alphabet for MLT3 is {A}={ -1, 0, +1}. In MLT3 transmission, a logical 1 is transmitted by either a -1 or a +1 while a logic 0 is transmitted as a 0. A transmission of two consecutive logic "1"s does not require the system to pass through zero in the transition. A transmission of the logical sequence ("1", "0", "1") would result in transmission of the symbols (+1, 0, -1) or (-1, 0, +1), depending on the symbols transmitted prior to this sequence. If the symbol transmitted immediately prior to the sequence was a +1, then the symbols (+1, 0, -1) are transmitted. If the symbol transmitted before this sequence was a -1, then the symbols (-1, 0, +1) are transmitted. If the symbol transmitted immediately before this sequence was a 0, then the first symbol of the sequence transmitted will be a +1 if the previous logical "1" was transmitted as a -1 and will be a -1 if the previous logical "1" was transmitted as a +1.

[0006] The detection system in the MLT3 standard needs to distinguish between 3 voltage levels, instead of two voltage levels in a more typical two level system. The signal to noise ratio required to achieve a particular bit error rate is~~ higher for MLT3 signaling than for two level systems.~~ The advantage of the MLT3 system, however, is that the energy spectrum of the emitted radiation from the MLT3 system is concentrated at lower frequencies and therefore

more easily meets FCC radiation emission standards for transmission over twisted pair cables. Other communication systems may use a symbol alphabet having more than two voltage levels in the physical layer in order to transmit multiple bits of data using each individual symbol.

[0007] In Gigabit Ethernet over twisted pair Category-5 cabling, for example, PAM-5 data can be transmitted over four twisted copper pairs at an individual twisted pair baud rate of 125 Mbaud. In PAM-5, data is sent with 5 voltage levels, designated as symbol alphabet {A}= {-2, -1, 0, 1, 2} although the actual voltage levels may be different from those. Each symbol, therefore, can be used to code more than one bit of data.

[0008] Any other modulation scheme for symbol coding can be utilized, including quadrature amplitude modulation (QAM). In QAM schemes, for example, the symbols are arranged on a two-dimensional (real and imaginary) symbol constellation (instead of the one-dimensional constellations of the PAM-5 and MLT-3 symbol alphabets).

[0009] There is a need for transmitters and receivers for receiving transmission over multiple twisted copper pairs using larger symbol alphabets (i.e., 3 or more symbols). There is also a need for transceiver (transmitter/receiver) systems that, while operating at high symbol rates, have low bit error rates.

SUMMARY

[0010] Accordingly, a receiver and detection method for receiving transmission of data over multiple wires using encoded data symbol schemes having multiple symbols is described. A receiver according to the present invention includes multiple detectors for detecting one symbol from each of the multiple wires simultaneously (i.e., multiple 1-D detectors), an equalizer coupled to each of the

detectors for equalizing the symbol stream from each of the multiple wires, and a multi-dimensional error analyzer/decoder for simultaneously making hard decisions regarding the symbol output of the symbols transmitted over each of the multiple wires.

[0011] Individual symbols can have any modulation scheme, including those with multi-dimensional constellations. The terminology of detecting N-dimensional (N-D) symbols refers to the number of individual symbols detected in each clock cycle, and not to the dimension of the symbol constellation.

[0012] Each of the equalizers can be any equalizer structure, including linear or decision feedback equalizers. In one embodiment of the invention, the equalizers include sequence detection equalizers. In another embodiment of the invention, the equalizers include simplified decision feedback equalizers.

[0013] In another embodiment, the equalizer includes a sequence detection equalizer in combination with a linear equalizer or a decision feedback equalizer. In some embodiments, the sequence detection equalizer is a reduced state sequence detector. In some embodiments, the decision feedback equalizer is a simplified decision feedback equalizer.

[0014] In general, a transceiver according to the present invention can utilize any symbol alphabet. In some embodiments, a PAM-5 symbol alphabet is utilized.

[0015] Some embodiments of the invention can also include an error analysis decoder. The receiver receives signals from N individual wires and, for each wire, includes a linear equalizer in combination with a one-dimensional (1-D) sequence detector. The N output signals from the N 1-D sequence detectors are input to a N-D decoder that makes a final decision on the N-D symbol. In some embodiments of the invention, the error analysis decoder operates with

lattice encoding schemes. In some other embodiments of the invention, the error analysis decoder operates with a parity encoding scheme.

[0016] These and other embodiments of a transceiver system according to the present invention are further described below with reference to the following figures.

BRIEF DESCRIPTION OF THE FIGURES

[0017] Figure 1 shows a block diagram of a transceiver having multiple transport channels.

[0018] Figure 2 shows a block diagram of an encoder capable of both trellis encoding and parity encoding.

[0019] Figure 3 shows a trellis diagram for state transitions among subsets of PAM-5 four-dimensional (4-D) symbols.

[0020] Figure 4 shows a block diagram of a model of transmission channel between a transmitter and a receiver system.

[0021] Figure 5A shows a block diagram of a N-D receiver system according to the present invention.

[0022] Figure 5B shows a block diagram of a receiver according to the present invention for the N-D receiver system of Figure 5A.

[0023] Figure 6 shows a block diagram of a linear equalizer.

[0024] Figure 7 shows a block diagram of the equalizer and decoder portions of a 4-D receiver system that includes linear equalizers and parity encoding.

[0025] Figure 8 shows a block diagram of a decision feedback equalizer.

[0026] Figure 9 shows a block diagram of the equalizer and decoder portions of a 4-D receiver system that includes decision feedback equalization.

[0027] Figure 10 shows a block diagram of a 4-D receiver system that includes sequence detection equalizers.

[0028] Figure 11A shows a block diagram of a sequence detector.

[0029] Figure 11B shows a block diagram of an embodiment of an equalizer system.

[0030] Figure 12 shows an example of a trellis diagram for state transitions between PAM-5 symbols.

[0031] Figure 13 shows a block diagram of an embodiment of a sequence detector equalizer with decision feedback according to the present invention.

[0032] Figure 14 shows a block diagram of an embodiment of a sequence detector equalizer having soft outputs according to the present invention.

[0033] Figure 15 shows a block diagram of an embodiment of a parity code receiver that includes soft output sequence detection equalizers according to the present invention.

[0034] Figure 16 shows an embodiment of a reduced complexity sequence detector having soft outputs according to the present invention.

[0035] Figure 17 shows an example of a trellis diagram illustrating state changes between reduced states of a reduced complexity sequence detector, including multiple branches between states.

[0036] Figure 18 shows a trellis diagram illustrating state changes between the reduced states of a reduced complexity sequence detector after branch path decisions have been made.

[0037] Figure 19 shows a block diagram of a decision feedback equalizer having a simplified feedback section according to the present invention.

[0038] Figure 20A shows a sequence detector in combination with a decision feedback equalizer.

[0039] Figure 20B shows an embodiment of the feedback section of the decision feedback equalizer shown in Figure 20A that includes a simplified feedback section.

[0040] In the figures, components having similar functions are often labeled similarly.

DETAILED DESCRIPTION

[0041] Figure 1 shows a block diagram of a transceiver system 100 according to the present invention. Transceiver system 100 includes transmitter 101, receiver 107, and transport wires 103-1 through 103-N. A digital data stream is input to transmitter 101 by a first host 150. Transmitter 101 includes encoder 102, which encodes the data for transmission over transport wires 103-1 through 103-N. Transmission coupler 112 couples the encoded symbols received from encoder 102 to transmission channel 104. Data is output by transmission coupler 112 as symbols $a_{k,1}$ through $a_{k,N}$ over wires 103-1 through 103-N (collectively referred to as cable 103), respectively, which are coupled between transmitter 101 and receiver system 107. Wires 103-1 through 103-N can be of any transport medium or combination of transport media. Transport media include, for example, category-5 twisted copper pair, optical fiber and coax cable.

[0042] The index k in the above notation indicates the kth time cycle of the data transmission. The indicator N is an integer indicating the number of individual transport wires in cable 103. In general, any number of wires can be included in transceiver system 100. For gigabit

transmission, N is usually four (4), indicating a four wire connection between transmitter 101 and receiver system 107. Wires 103-1 through 103-4 are often twisted copper pair.

[0043] Transmission channel 104 collectively represents cable 103 and any distortion of the signals that occurs between transmitter 101 and a receiver 107. Each of wires 103-1 through 103-N, along with output couplers of coupler 112 and input couplers of detector 108, distorts signals as indicated by the associated transmission channel 104-1 through 104-N, respectively. The signals through wires 103-1 through 103-N are distorted by channel functions $f_1(Z)$ through $f_N(Z)$, respectively, and additionally suffer from a random noise addition $n_{k,1}$ through $n_{k,N}$, respectively. Receiver 107 includes signal detector 108 and a decoder 109 and ultimately outputs a data stream which corresponds to the data stream entering transmitter 101.

[0044] In Gigabit Ethernet, transmission can be conducted on four twisted copper pairs (i.e., wires 103-1 through 103-4 in Figure 1) in a full duplex fashion to achieve one gigabit per second throughput. *See* IEEE 802.3ab, Gigabit Long Haul Copper Physical Layer Standards Committee, 1997 (hereinafter "Gigabit Standard"). Transmitter 101 is coupled to first host 150 to receive data for transmission over transmission channel 104 to a second host 151 coupled to receiver 107. In general, first host 150 is further coupled to a receiver 111 for receiving data from transmission channel 104 and second host 151 is coupled to a transmitter 110 for sending data to transmission channel 104.

[0045] Although, in general, the detection system as described here is applicable to any scheme of data transmission (i.e., any symbol alphabet) over any number of transport wires (e.g., twisted copper pair), for exemplary purposes many of the examples below specifically describe a PAM-5 symbol alphabet transmitted over four twisted copper

pairs, as would be used in Gigabit Ethernet transmission over Category-5 twisted pair cabling. It should be recognized that other symbol alphabets and numbers of conductors can also be used and that one skilled in the art will recognize from this disclosure embodiments appropriate for other modulation schemes.

Encoder

[0046] According to the developed IEEE 802.3ab standard for Gigabit Ethernet transmission, the transmitted symbols on each of four conductors 103-1 through 103-4 are chosen from a five level Pulse Amplitude Modulation (PAM-5) constellation, with alphabet {A}={-2, -1, 0, +1, +2}. *See* Gigabit Standard. At each clock cycle, a single one-dimensional (1-D) symbol is transmitted on each wire. The four 1-D symbols, one on each of conductors 103-1 through 103-4, transmitted at a particular sample time k are considered to be a single 4-D symbol. In general, data entering transmitter 101 can be encoded into a N-D symbol, instead of into N 1-D symbols, by encoder 102. Encoder 102 may be any type of N-D encoder, including a N-D parity code encoder and a N-D trellis code encoder.

[0047] To achieve one gigabit per second communications, a Gigabit Ethernet transceiver needs to achieve a throughput of 250 Megabits per second over each of four transport wires 103-1 through 103-4. Therefore, at a 125 MHz baud rate, two bits must be transmitted at each sample time across each wire of cable 103. Although a PAM-5 system is applicable, a four level PAM system, for example, does not provide redundancy which allows for error correction coding necessary to achieve a bit error rate (BER) of 10^{-10}, as required by the Gigabit Ethernet standard. *See* Gigabit Standard.

[0048] In addition, extra channel symbols are needed to represent Ethernet control characters. Therefore, five

level PAM (PAM-5) with either a parity check code or trellis coding is often utilized in Gigabit Ethernet transmission. According to the Gigabit Standard, the trellis code is the only coding utilized. Alphabets having more than five level 1-D symbols may also be utilized for gigabit transmission while achieving the required BER.

[0049] At 125 Mbaud, each 4-D symbol needs to transmit at least eight bits. Therefore, 256 different 4-D symbols plus those required for control characters are required. By transmitting a 4-D PAM-5 symbol alphabet, there are $5^4 = 625$ possible symbols. This number of symbols allows for 100% redundancy in the data as well as for several control codes. Symbol alphabets having more than five symbols yield even greater redundancy. However, because of the difficulties in distinguishing between the higher number of voltage levels, higher error rates may be suffered in receiver detection systems, such as detector 108 of Figure 1.

[0050] The Gigabit Ethernet standard (*see* Gigabit Standard) allows a trellis code. However, both a 4-D eight state trellis code and a 4-D parity code have been proposed. The trellis code achieves 6dB of coding gain over uncoded PAM-5 while the parity code achieves 3dB of coding gain. While encoding, the choice between encoders can be encoded in a TX_CODING bit, which can be set to one for trellis coding. One skilled in the art will recognize that embodiments of the present invention are applicable to any error correction technique.

4-D Trellis Encoding

[0051] Trellis encoding in a conventional 4-D eight state code is described in Part II, Table IV of G. Ungerboeck, "Trellis-Coded Modulation with Redundant Signal Sets, Part I: Introduction", *IEEE Communications Mag.*, vol. 25, no. 2, pp. 5 - 11, February 1987, and "Trellis-Coded Modulation with Redundant Signal Sets, Part II: State of the Art", *IEEE*

Communications Mag., vol. 25, no. 2, pp. 12 - 21, February 1987 (hereinafter collectively "Ungerboeck"). An embodiment of an eight state trellis encoder 200 similar to that described in Ungerboeck is shown in Figure 2. Trellis encoder 200 receives eight bits, bits 0 through 7, and outputs 9 bits, bits 0 through 7 plus a parity bit, provided that TX_CODING is set to 1. The parity bit is produced from a rate 2/3 memory 3, systematic convolutional encoder 201.

[0052] Convolutional encoder 201 includes delays 202, 204 and 206, each of which delays its input signal by one clock cycle. The output signal from delay 202 is XORed (exclusive-ORed) with bit 6 in XOR 203 and input to delay 204. The output signal from delay 204 is XORed with bit 7 in XOR 205 and input to delay 206. The output signal from delay 206 is input to AND gate 207 and input to delay 202. AND gate 207 generates the parity bit, which is one (1) if TX_CODING is one (1) and the output signal from delay 206 is 1, and zero otherwise. In embodiments that only utilize trellis coding (i.e., TX_CODING is always 1), AND gate 207 may be excluded from encoder 200 and the parity bit is the output signal from convolutional encoder 201.

[0053] The two input bits to convolutional encoder 201 (bits 6 and 7), and the parity bit produced by encoder 201 select one of eight subsets, D0 through D7, of the 4-D symbol alphabet. The nine output bits are mapped to one 4-D PAM-5 symbol that is then transmitted across cable 103 (Figure 1) as four 1-D PAM-5 symbols.

[0054] Each 1-D PAM-5 symbol is a member of one of two families, X (odd) and Y (even). The odd set X contains the PAM-5 symbols {-1, +1} and the even set Y contains the PAM-5 symbols {-2, 0, +2}. Table 1 shows a definition of the eight subsets of 4-D symbols labeled D0 through D7. The definition is based on the membership of 1-D PAM-5 symbols that are represented in each subset. Table 1 also shows the number of 4-D symbols included in each of the subsets.

[0055] The notation describing each subset indicates the families of 1-D PAM-5 symbols on each of the four wires describing membership in that subset. The notation XXYX, for example, indicates a set of four 1-D PAM-5 symbols where the first symbol is from type X, the second symbol is from type X, the third symbol is from type Y, and the fourth symbol is from type X. Therefore, conductor 103-1 (Figure 1) carriers a symbol of type X, conductor 103-2 carries a symbol of type X, conductor 103-3 carries a symbol of type Y and conductor 103-4 carries a symbol of type X.

Table 1 -- Mapping of 4-D symbols into sets D0 through D7

Subset	X Primary Code	Y Primary Code	# of Symbols
D0	XXXX	YYYY	97
D1	XXXY	YYYX	78
D2	XXYY	YYXX	72
D3	XXYX	YYXY	78
D4	XYYX	YXXY	72
D5	XYYY	YXXX	78
D6	XYXY	YXYX	72
D7	XYXX	YXYY	78

[0056] The parity bit and bits 6 and 7 are input to set select 210 (Figure 2). Set select 210 determines a particular subset selection p, indicating that subset D_P of subsets D_0 through D_7 has been selected. In one embodiment, subset selection p is determined according to the formula

$$p = 4*BIT6 + 2*BIT7 + 1*Parity, \qquad (1)$$

where BIT6 is bit 6, BIT7 is bit 7, and Parity is the parity bit.

[0057] A point within subset p is chosen by the six least significant bits of the input, bits 0 through 5. Bits 0 through 5 are input to a 4-D PAM-5 mapper 211 along with the

output χ of set select 210. 4-D PAM-5 mapper 211 determines the 4-D PAM-5 symbol within set D_p which represents the eight input bits, bits 0 through 7, and the parity bit. Each subset, D0 through D7, contains more than the 64 points required to encode the six bits, bits 0 through 5. These additional points are either used as control characters or not used at all.

[0058] The subset mapping shown in Table 1 is chosen such that the squared Euclidean distance between any pair of points in the same subset is greater than or equal to four (4). For example, the squared distance between two points in subset D0 can be expressed as $(X_{11}-X_{12})**2$ + $(X_{21}-X_{22})**2$ + $(X_{31}-X_{32})**2$ + $(X_{41}-X_{42})**2$. Because X_{ij} = +2, 0, or -2, the shortest squared distance is (2-0)**2 = 4, which occurs when only one value differs between the two points. In addition, the squared Euclidean distance between points in different even subsets is greater than or equal to 2 and the squared Euclidean distance between points in different odd subsets is greater than or equal to 2. As also can be seen from Table 1, even numbered subsets (D_0, D_2, D_4, D_6) have an even number of symbols of type X and an even number of symbols from type Y while odd numbered subsets (D_1, D_3, D_5, D_7) have an odd number of symbols of type X and an odd number of symbols of type Y.

[0059] Figure 3 shows a trellis diagram resulting from the convolutional encoder shown in Figure 2. The states of the trellis in Figure 3, S_0 through S_7, are determined by the bits in delay elements 202, 204 and 206. The state S_q is determined by q=4*(the output signal from delay 202)+2*(the output signal from delay 204) + 1*(the output bit from delay 206). The subsets D_p are determined by the transition from one state to another as shown in Figure 3. The subsets D_p are given in Table 1.

[0060] As shown in Figure 3, all transitions out of a particular state are either all even subsets (D_0, D_2, D_4, and

D_6) or all odd subsets (D_1, D_3, D_5, and D_7). Similarly, the transitions into a particular state are either all even subsets or all odd subsets. The minimum squared distance between outgoing transitions, therefore, is equal to two (2) and the minimum squared distance between incoming transitions is also two (2).

[0061] As a result, the minimum squared distance between valid sequences is greater than or equal to 4, which can be seen from the fact that any two paths that originate from the same node and end at the same node, but diverge at some point in the middle, must then converge at a later time. Both the divergence and the convergence have a squared distance of at least 2, thus the total squared distance must be at least 4. In the case where these two paths do not diverge, then they must contain different points from the same subset. Because the minimum squared distance between points in the same subset is equal to 4, the minimum squared distance between valid sequences is 4. As is conventionally known, a coding gain of 6dB with respect to uncoded PAM-5 constellations is therefore experienced.

4-D Parity Code

[0062] The 4-D parity code can also be transmitted using the same encoder, encoder 200 of Figure 2. To do so, the TX_CODING bit is set to zero. As a result, the parity bit produced by convolutional encoder 201 is always 0. Therefore, set select 211 always chooses even subsets (i.e., X=0, 2, 4 or 6). The minimum squared Euclidean distance between any two 4-D symbols, therefore, is 2 and a coding gain of 3dB is experienced over uncoded PAM-5.

Transmitter Coupler

[0063] Transmitter coupler 112 (Figure 1) couples the N-D symbol from encoder 102 to transmission channel 104. In some embodiments, coupler 112 can include pre-coding of

signals to each of lines 103-1 through 103-N. Some pre-coding may at least partially overcome the expected intersymbol interference effects experienced in transmission channels 104-1 through 104-N, respectively. However, some pre-coding, such as partial response shaping, for example, may actually add to the intersymbol interference effects.

[0064] Coupler 112 converts symbols received from encoder 102 to appropriate voltages for transmission to receiver 107 through transmission channel 104. In many embodiments of PAM-5 transmission, for example, those voltages are (-1, -1/2, 0, +1/2, +1) volts corresponding to the PAM-5 symbols (-2, -1, 0, 1, 2), respectively.

Transmission Channel Characteristics

[0065] An input symbol stream $\{a_{k,1}\}$ through $\{a_{k,N}\}$ is input to each of wires 103-1 through 103-N, respectively, of transmission channel 104 (Figure 1) by transmitter 101. Wires 103-1 through 103-N can be twisted copper pair, or some other transmission medium such as coaxial cable or optical fiber. Transmission channel 104 represents the effects that wires 103-1 through 103-N, transmission coupler 112 and receiver couplers in detector 108 have on the input symbol streams $\{a_{k,1}\}$ through $\{a_{k,N}\}$ in transmission between transmitter 101 and receiver 107. Each transmission channel 104-1 through 104-N includes a corresponding one of wires 103-1 through 103-N, respectively, for the symbol stream $\{a_{k,1}\}$ through $\{a_{k,N},\}$ respectively.

[0066] Figure 4 shows a single transmission channel 104-L that represents the Lth one of transmission channels 104-1 through 104-N, where L can be any one of 1 through N. The symbol $a_{k,L}$, representing the symbol transmitted on wire 103-L in the kth time period, is transmitted through transmission channel 104-L. The symbol stream $\{a_{k,L}\}$ of Figure 4 can be NRZ, MLT3, PAM-5 or any other symbol alphabet and modulation. The transmitted symbols in the

sequence $\{a_{k,L}\}$ are members of the symbol alphabet {A}. In the case of five-level PAM signaling, the symbol alphabet {A} is given by {-2, -1, 0 +1, +2}.

[0067] The channel response 105-L is represented by the channel function $f_L(Z)$. In Figure 1, each of conductors 103-1 through 103-N can experience a different channel function $f_1(Z)$ through $f_N(Z)$, respectively. The addition of random noise $n_{k,L}$ in Figure 4 is represented by adder 106-L. Again, in Figure 1 each of conductors 103-1 through 103-N experiences a different random noise $n_{k,1}$ through $n_{k,N}$, respectively. The signal $x_{k,L}$ suffering from channel distortion, random noise, and a flat signal loss is received by receiver 107.

[0068] For the sake of simplicity, a baseband transmission system is assumed, although the techniques shown are easily extended to a passband transmission system. (*See* E.A. LEE AND D.G. MESSERCHMITT, DIGITAL COMMUNICATIONS (1988).) It is also assumed that the channel model includes the effects of transmit and receive filtering. In addition, the transmission channel is assumed to be linear in that two overlapping signals simply add as a linear superposition. The Z-transform (*see* A. V. OPPENHEIM & R. W. SCHAFER, DISCRETE-TIME SIGNAL PROCESSING (1989)) of the sampled transmission channel 104-L shown in Figure 4 is given by the channel function polynomial

$$f_L(Z)=f_{0,L} + f_{1,L}Z^{-1} + \ . \ . \ . \ f_{j,L}Z^{-j} + \ . \ . \ . \ + f_{R,L}Z^{-R}, \qquad (2)$$

where $f_{0,L}$, ..., $f_{j,L}$, ..., $f_{R,L}$ are the polynomial coefficients. The coefficient $f_{j,L}$ represents the dispersed component of the (k-j)th symbol present in the $a_{k,L}$th symbol and R is a cut-off integer such that $f_{j,L}$ for j>R is negligible. The polynomial $f_L(Z)$ represents the Z-transformation of the frequency response of the transmission

channel (Z^{-1} represents a one period delay). (*See* A. V. OPPENHEIM & R. W. SCHAFER, DISCRETE-TIME SIGNAL PROCESSING (1989).)

[0069] The noiseless output of the channel at sample time k is given by

$$r_{k,L} = f_{0,L} * a_{k,L} + f_{1,L} * a_{k-1,L} + \ldots f_{R,L} * a_{k-R,L}, \quad (3)$$

where, without loss of generality, $f_{0,L}$ can be assumed to be 1. Thus, the channel output signal at time k depends, not only on transmitted data at time k, but on R past values of the transmitted data. This effect is known as "intersymbol interference" (ISI). (*See* LEE & MESSERSCHMITT.) The ISI length of the transmission channel defined by Equations 2 and 3 is R, the number of past symbols contributing to ISI.

[0070] Intersymbol interference is a result of the dispersive nature of the communication channel. The IEEE LAN standards require that systems be capable of transmitting and receiving data through at least a 100 meter cable. In many instances, a single symbol may affect symbols throughout the transmission cable.

[0071] The noise element of the input signal is represented by the sequence $\{n_{k,L}\}$. Therefore, the noisy output signal from transmission channel 104-L is given by

$$x_{k,L}=r_{k,L} + n_{k,L}, \quad (4)$$

where the noise samples $\{n_{k,L}\}$ are assumed to be independent and identically distributed Gaussian random variables (*see* LEE & MESSERSCHMITT) with variance equal to σ^2.

Receiver

[0072] Figure 5A shows a block diagram of an embodiment of a baseband receiver system 500 according to the present

invention. Other receiver systems may also utilize aspects of the present invention. Embodiments of receiver system 500 may include any number of transport wires 103-1 through 103-N (with associated transmission channels 104-1 through 104-N, respectively) carrying input signal streams $x_{k,1}$ through $x_{k,N}$, respectively.

[0073] Receiver system 500 includes receivers 501-1 through 501-N, one for each of lines 103-1 through 103-N, respectively. Receiver 501-j, an arbitrarily chosen one of receivers 501-1 through 501-N, includes filter/digitizer 502-j, equalizer 505-j, and coefficient update 506-j. Signal $x_{k,j}$ from wire 103-j is received by filter/digitizer 502-j. Filter/digitizer 502-j filters, digitizes and amplifies the signal $x_{k,j}$ and outputs a signal $y_{k,j}$. Equalizer 505-j receives the signal $y_{k,j}$, equalizes it to remove the effects of intersymbol interference, and outputs a signal $a'_{k,j}$, which is the output signal for receiver 501-j. Filter/digitizer 502-j can be arranged to include filters that partially remove the ISI from signal $x_{k,j}$ before digitizing the signal. *See*, e.g., U.S. Patent Application 09/561,086, Manickam et al., filed on the same date as the present application, assigned to the same assignee as the present application, herein incorporated by reference in its entirety.

[0074] Coefficient update 506-j inputs decided-on symbols $\hat{a}_{k,j}$ and other parameters and adaptively chooses parameters for filter/digitizer 502-j and equalizer 505-j (e.g., amplifier gain, multiplier coefficients, filter parameters, echo cancellation, NEXT cancellation, and timing parameters).

[0075] One skilled in the art will recognize that each of receivers 501-1 through 501-N can be different. That is, each of filters/digitizers 502-1 through 502-N and equalizers 505-1 through 505-N can be individually matched

to receive input signals from the corresponding one of wires 103-1 through 103-N.

[0076] Figure 5B shows a representative example of receiver 501-j. Receiver 501-j is one of receivers 501-1 through 501-N. Receiver 501-j includes filter/digitizer (or receiver/digitizer) 502-j in series with equalizer 505-j. Receiver/digitizer 502-j includes, in series, filter/echo canceller 508-j, analog-to-digital converter (ADC) 509-j, and amplifier 510-j. One skilled in the art will recognize that the order of these components can be altered from that shown in Fig. 5B. For example, amplifier 510-j can be implemented before filter/echo canceller (or simply filter) 508-j, or filter 508-j can be implemented, completely or partially, digitally after ADC 509-j.

[0077] Parameters to control the components of receiver 501-j can be adaptively chosen by coefficient update 506-j. Coefficient update 506-j adaptively determines the equalizer coefficients of equalizer 505-j, the gain g_j of amplifier 510-j, the timing coefficient τ_j of ADC 509-j, and filter coefficients for filter 508-j. In some embodiments, coefficient update 506-j can calculate a baseline wander correction signal w_j which is subtracted from the output sample of ADC 509-j at baseline wander correction adder 511-j. Baseline wander correction is discussed in U.S. Patent Application 09/151,525, filed September 11, 1998, Raghavan, assigned to the same assignee as the present application, now U.S. Patent 6,415,003, herein incorporated by reference in its entirety.

[0078] Some embodiments of receiver 501-j include a cable quality and length indicator 512-j that indicates a wire length L and wire quality Q.

[0079] Echo noise, which is a result of impedance mismatches in the duplex link causing some of the transmitted signal energy to be reflected back into a

receiver, and near end crosstalk (NEXT) noise, which is caused by the interference from a transmitter, i.e., transmitter 110 (Figure 1), physically located adjacent to receiver 107, can be canceled through adaptive algorithms in filter 508-j. The cancellation of echo noise and NEXT noise is nearly complete because receiver system 500 has access to the data transmitted by an adjacent transmitter (transmitter 110 in Figure 1). Coefficient update 506-j, therefore, may input parameters from a decoder/slicer, host 151 or other controller in order to adjust filter 508-j to cancel echo and NEXT noise. Note that, in Figure 1, transmitter 101 may also be adjacent to an accompanying receiver 111.

[0080] Filter 508-j can also include an anti-aliasing filter. An anti-aliasing filter prevents aliasing by passing the input signal, received from wire 103-j, through a low pass filter to reject out-of-band noise. As such, any conventional anti-aliasing filter can be utilized as an anti-aliasing filter portion of filter 508-j.

[0081] The analog-to-digital converter (ADC) 509-j samples and holds the input signal for a duration of the symbol period T, which in one embodiment of the invention is 8 ns. In general, embodiments of the invention can utilize any symbol period. Techniques for analog-to-digital conversion that can be used in ADC 509-j are well known. The digitized signals in the receiver are interchangeably referred to in this disclosure as samples or signals.

[0082] Amplifier 510-j amplifies the samples received from wire 103-j through transmission channel 104-j in order to correct for signal loss during transmission. The gain g_j of amplifier 510-j can be adaptively chosen by coefficient update 506-j in order to optimize the operation of receiver 501-j. One of ordinary skill in the art will recognize that digital amplifier 510-j can be located anywhere in receiver 501-j between ADC 509-j and equalizer 505-j. In general,

amplifier 510-j can also be an analog amplifier located anywhere between input channel 104-j and ADC 509-j.

[0083] In some embodiments, coefficient update 506-j can also calculate the length and quality of wire 103-j. Cable length and quality determination is discussed in U.S. Patent Application 09/161,346, filed September 25, 1998, Raghavan et al., assigned to the same assignee as the present application, now U.S. Patent 6,438,163, herein incorporated by reference in its entirety.

[0084] The output sample $y_{k,j}$ from receiver/digitizer 502-j is input to equalizer 505-j. Equalizer 505-j can be any kind of equalizer structure. Types of equalizer structures include linear equalizers, decision feedback equalizers, and sequence detection equalizers. Equalizers of these types for 100 or 1000 BASE-T Ethernet over category-5 wiring, 24 gauge twisted copper pair, are described in U.S. Patent Application 08/974,450, filed November 20, 1997, Raghavan, assigned to the same assignee as the present application, now U.S. Patent 6,038,269, herein incorporated by reference in its entirety; and U.S. Patent Application 09/020,628, filed February 9, 1998, Raghavan, assigned to the same assignee as the present application, now U.S. Patent 6,115,418, herein incorporated by reference in its entirety.

[0085] Additionally, receivers of the type described above as receivers 501-1 through 501-N are further described in U.S. Patent Applications 09/151,525 and 09/161,346, both cited above.

[0086] In receiver system 500 of Figure 5A, the output samples $a'_{k,1}$ through $a'_{k,N}$ from receivers 501-1 through 501-N, respectively, are input to decoder 507. Decoder 507 decides on a N-D symbol based on the samples from receivers 501-1 through 501-N. Each of the N 1-D symbols of the N-D symbol is the result of the N-D symbol pick in decoder 507.

[0087] Figure 6 shows a block diagram of a linear equalizer 600 having R+1 multipliers, where R is any integer greater than or equal to 0. Linear equalizer 600 includes delays 601-1 through 601-R connected in series. The output samples from delays 601-1 through 601-R are also input to multipliers 602-1 through 602-R, respectively. The input sample y_k to equalizer 600 is input to multiplier 602-0. The input samples to multipliers 602-0 through 602-R are multiplied by the corresponding one of multiplier coefficients C_0 through C_R, respectively, and summed in adder 603. Multiplier coefficients C_0 through C_R can be adaptively chosen to optimize the performance of equalizer 600. In Figure 5B, for example, coefficient update 506-j adaptively chooses equalizer parameters for equalizer 505-j.

[0088] Equalizer 600 (Figure 6) executes the transfer function

$$T = C_o + C_1 Z^{-1} + \cdots + C_j Z^{-j} + \cdots + C_R Z^{-R}. \tag{6}$$

For simplicity, the wire designation L has been neglected. It is understood that each transmission channel includes a separated equalizer, each of which can be a unit of equalizer 600 having its own transfer function T.

[0089] In a zero-forcing linear equalizer (ZFLE), the transfer function T is the inverse of the frequency response of the channel f(Z) (see Equation 2 with the wire designation L neglected). In a minimum mean squared error based linear equalizer (MMSE-LE), the transfer function is arranged to optimize the mean squared error between the transmitted data signal and the detected data symbols. A compromise, then, is found between the un-canceled ISI and the noise variance at the output terminal of the equalizer. (*See* B. SKLAR, DIGITAL COMMUNICATIONS, FUNDAMENTALS AND APPLICATIONS (PTR Prentice Hall, Englewood Cliffs, NJ, 1988).)

[0090] The output sample from linear equalizer 600, executing transfer function T, is given by

$$a'_k = C_0 y_k + C_1 y_{k-1} + \cdots + C_j y_{k-j} + \cdots + C_R y_{k-R}\,, \tag{7}$$

where C_0 through C_R are the equalizer coefficients, y_{k-j} is the input signal to equalizer 600 during the time period that is j periods before the kth period, and k represents the current time period.

[0091] The output sample a'_k from equalizer 600 is usually input to a slicer 604 which decides, based on its input sample a'_k, what symbol $\hat{a}_k$ was transmitted during time period k. The symbol $\hat{a}_k$ is chosen from the symbol alphabet used for transferring data that is closest to input signal a'_k.

[0092] A linear equalizer can be implemented using either parity coding or trellis coding systems. When linear equalization is used with parity coding, a separate linear equalizer is used on each transport wire. In Figure 5A, for example, each of equalizers 505-1 through 505-N can include a linear equalizer.

[0093] As an example, Figure 7 shows a portion of a 4-D receiver system 710, which includes 4-D decoder 700 and linear equalizers 600-1 through 600-4. Linear equalizers 600-1 through 600-4 provides equalization for each of channels 104-1 through 104-4, respectively. Decoder 700 can be any type of decoder including a parity code decoder and a 4-D trellis decoder. As an example, decoder 700 is shown as a parity code decoder. Also, although a 4-D decoder is shown in Figure 7, it is understood that in general receiver system 710 can have any number of parallel input signals.

[0094] The input samples to linear equalizers 600-1 through 600-4 are the output samples $y_{k,1}$ through $y_{k,4}$ from receivers/digitizers 502-1 through 502-4 (Figure 5B),

respectively. The subscripts 1-4 reference each of the four wires 103-1 through 103-4, respectively. The output samples $a'_{k,1}$ through $a'_{k,4}$ from linear equalizers 600-1 through 600-4, respectively, are input to slicers 604-1 through 604-4, respectively. In Figure 7, slicers 604-1 through 604-4 are 1-D slicers. If PAM-5 symbol coding is utilized, then slicers 604-1 through 604-4 are 1-D PAM-5 slicers that each output the PAM-5 symbol closest to input sample $a'_{k,1}$ through $a'_{k,4}$, respectively.

[0095] The output symbols $\hat{a}_{k,1}$ through $\hat{a}_{k,4}$ from slicers 604-1 through 604-4, respectively, are input to parity check 702. Additionally, each of samples $a'_{k,1}$ through $a'_{k,4}$ is subtracted from the corresponding one of symbols $\hat{a}_{k,1}$ through $\hat{a}_{k,4}$, respectively, in adders 701-1 through 701-4, respectively, to calculate errors $e_{k,1}$ through $e_{k,4}$, respectively. In general, the error signal $e_{k,i}$, where i is 1 through 4, is then given by

$$e_{k,i}=a'_{k,i}-\hat{a}_{k,i} \quad (8)$$

[0096] The parity of the 4-D symbol that results from the four 1-D symbols is checked in parity check 702. In parity coding, the 4-D symbol is chosen from an even subset of all 4-D symbols. Therefore, if PAM-5 coding is used, there are an even number of PAM-5 symbols from family X (odd symbols) and an even number from family Y (even symbols). Parity check 702 checks the parity of the four input symbols, $\hat{a}_{k,1}$ through $\hat{a}_{k,4}$, by determining whether the sum of the four symbols is even or not. An odd parity indicates that there is an error in at least one of the decided symbols $\hat{a}_{k,1}$ through $\hat{a}_{k,4}$. The result of the parity check is input to final decoder 704.

[0097] The calculated error signals $e_{k,1}$ through $e_{k,4}$ are input to error analysis 703. Error analysis 703 determines

which of the four error signals $e_{k,1}$ through $e_{k,4}$ is greatest and the sign of that error. Error analysis 703 outputs a sign signal Sgn and an identifier W for the symbol having the greatest error.

[0098] Final decoder 704 inputs the parity signal from parity check 702, the four 1-D PAM-5 symbols $\hat{a}_{k,1}$ through $\hat{a}_{k,4}$ from slicers 604-1 through 604-4, respectively, the identifier W for the symbol having the greatest error and the sign Sgn of that error and outputs the PAM-5 symbols $\hat{a}'_{k,1}$ through $\hat{a}'_{k,4}$ in response. If the parity is even, the 4-D symbol defined by 1-D symbols $\hat{a}'_{k,1}$ through $\hat{a}'_{k,4}$ is then given by $\hat{a}_{k,1}$ through $\hat{a}_{k,4}$, respectively. The parity coding scheme, therefore, will pass erroneous 4-D symbols containing simultaneous errors in two of the 1-D symbols.

[0099] If the parity is odd, however, the results of error analysis 703 are used to correct the output symbols. Because the symbol having the greatest error is the one that is most likely incorrect, the value of the symbol indicated by identifier W is corrected by either increasing or decreasing that symbol by one symbol in the symbol alphabet in response to the sign Sgn of the error (in this example, increased for a positive sign and decreased for a negative sign). The new set of four symbols, the three uncorrected symbols having the lowest error and the corrected symbol, is output as the 4-D symbol defined by 1-D symbols $\hat{a}'_{k,1}$ through $\hat{a}'_{k,4}$.

[0100] Although simple to implement, the primary disadvantage of a linear equalizer is that, while removing the ISI from the input signal, it may cause the random noise to be enhanced. This is especially true in twisted copper pair channels where the frequency response of the channel has significant attenuation across the transmitted signal bandwidth. Hence, in twisted-pair channels, such as is used with Gigabit Ethernet, linear equalization often does not perform well enough to be practical.

[0101] Figure 8 shows a decision feedback equalizer 800. Decision feedback equalizer 800 includes a feedforward section 810, a feedback section 811 and an adder 804. Feedforward section 810 includes delays 801-1 through 801-M coupled in series, multipliers 802-1 through 802-M coupled to receive the output signals from delays 801-1 through 801-M, respectively, multiplier 802-0 coupled to receive the input sample to decision feedback equalizer 800, and an adder 803 coupled to receive the output signals from each of multipliers 802-0 through 802-M. The output signal from each of multipliers 802-0 through 802-M is the input signal to that multiplier multiplied by the corresponding one of feedforward multiplier coefficients C_0 through C_M. Feedforward multiplier coefficients C_0 through C_M can be adaptively chosen in order to optimize the performance of the equalizer. In Figure 5B, for example, coefficient update 506-j adaptively chooses equalizer coefficients to optimize the performance of receiver 501-j.

[0102] Adder 803 (Figure 8) sums the output signals from each of multipliers 802-0 through 802-M. Feedforward section 810, therefore, executes the transfer function

$$T_{FF} = C_o + C_1 Z^{-1} + \cdots + C_j Z^{-j} + \cdots + C_M Z^{-M} . \tag{9}$$

The output signal from feedforward section 810, therefore, is given by

$$a'_k = C_0 y_k + C_1 y_{k-1} + \cdots + C_j y_{k-j} + \cdots + C_M y_{k-M} , \tag{10}$$

where y_{k-j} is the input sample to equalizer 800 during the (k-j)th time period. Again, the wire designation has been neglected for simplicity and it is understood that each of receivers 501-1 through 501-N (Figure 5) includes an

equalizer 505-1 through 505-N, respectively, any of which can be a decision feedback equalizer 800 (Figure 8).

[0103] One embodiment of feedback section 811 includes delays 805-1 through 805-P coupled in series. Multipliers 806-1 through 806-P are coupled to receive the corresponding output signals from delays 805-1 through 805-P, respectively. Multipliers 806-1 through 806-P multiply each of their input signals by the corresponding one of feedback multiplier coefficients B_1 through B_P, respectively. Feedback coefficients B_1 through B_P also can be adaptively chosen. Adder 807 sums the output signals from each of multipliers 806-1 through 806-P. Feedback section 811, therefore, executes the transfer function

$$T_{FB} = B_1 Z^{-1} + B_2 Z^{-2} + \cdots + B_P Z^{-P} . \tag{11}$$

The input signal to feedback section 811 is symbol $\hat{a}_k$. Consequently, the output signal from feedback section 811 is given by

$$a_k'' = B_1 \hat{a}_{k-1} + B_2 \hat{a}_{k-2} + \cdots + B_N \hat{a}_{k-P} . \tag{12}$$

[0104] A second embodiment of feedback section 811, which is later discussed in this disclosure, is shown as feedback section 1905 in Figure 19 and includes look-up table 1906.

[0105] The output signal a''_k from feedback section 811 is subtracted from the output signal a'_k from feedforward section 810 in adder 804. The input signal to slicer 808, then, is given by

$$a_k''' = a_k' - a_k'' . \tag{13}$$

Slicer 808 outputs the symbol $\hat{a}_k$ that is closest to the input signal a'''_k.

[0106] A decision feedback equalizer operates on the principle that if the past transmitted data is correctly detected, the ISI effects of these past data symbols can then be canceled from the currently received sample. As such, feedforward section 810 often contains no multipliers (i.e., $C_0 = 1$ and all other coefficients are 0) and output sample a'_k equals input sample y_k.

[0107] Past detected data samples contain no noise and therefore decision feedback equalizers do not suffer from noise enhancement. However, decision feedback equalizers do suffer from the effects of error propagation. If slicer 808 erroneously decides a previous symbol, the error will be propagated into subsequent decisions.

[0108] Figure 9 shows the equalization and decoder sections (*see*, for example, equalizers 505-1 through 505-N and decoder 507 of Figure 5A for the case in which N is 4) of a 4-D receiver 910. The equalization is accomplished using four decision feedback equalizers, one for each of the four input channels shown. Again, one skilled in the art will recognize that a receiver can have any number of input channels and that a 4-D receiver is shown here for example only. In Figure 9, receiver 910 includes a 4-D decoder 900. Although decoder 900 is shown as a parity code decoder, it is understood that decoder 900 can be any 4-D decoder. One skilled in the art will recognize that 4-D decoder 900 must correct symbols within a single clock cycle so that the results can be fed back through feedback sections (or taps) 811-1 through 811-4 to correct symbols $a'_{k,1}$ through $a'_{k,4}$, respectively.

[0109] The equalization for channel 104-1, for example, is accomplished by feedforward section (or tap) 810-1, feedback section 811-1, and adder 804-1. Similarly, channels 104-2 through 104-4, each includes the corresponding ones of

feedforward sections (or taps) 810-2 through 810-4, respectively, feedback sections 811-2 through 811-4, respectively, and adders 804-2 through 804-4, respectively.

[0110] Samples $y_{k,1}$ through $y_{k,4}$ originate from receiver input signals received from channels 104-1 through 104-4, respectively. Samples $y_{k,1}$ through $y_{k,4}$ are received into feedforward taps 810-1 through 810-4, respectively. The symbol outputs $\hat{a}'_{k,1}$ through $\hat{a}'_{k,4}$ from decoder 900, corresponding to a single 4-D symbol, are input to feedback taps 811-1 through 811-4, respectively. As was discussed with relation to Figure 8, the output signal $a''_{k,1}$ through $a''_{k,4}$ from each of feedback sections 811-1 through 811-4, respectively, is subtracted from the corresponding one of the output signals $a'_{k,1}$ through $a'_{k,4}$ from feedforward sections 810-1 through 810-4, respectively, in each of adders 804-1 through 804-4, respectively. The output signals $a'''_{k,1}$ through $a'''_{k,4}$ from adders 804-1 through 804-4, respectively, are inputted to decoder 900.

[0111] In an embodiment where decoder 900 is a 4-D parity code decoder, the input signals $a'''_{k,1}$ through $a'''_{k,4}$ to decoder 900 are received by slicers 901-1 through 901-4, respectively. Each of slicers 901-1 through 901-4 decides on an output symbol $\hat{a}_{k,1}$ through $\hat{a}_{k,4}$, respectively, based on its corresponding input signal $a'''_{k,1}$ through $a'''_{k,4}$, respectively. The output symbols from slicers 901-1 through 901-4 are inputted to parity check 903 and final decoder 905.

[0112] Parity check 903 sums the 1-D symbols $\hat{a}_{k,1}$ through $\hat{a}_{k,4}$ and determines whether the 4-D symbol is of even parity or odd parity. In parity coding using PAM-5 symbols, the 4-D symbol is chosen from an even subset and therefore each 4-D symbol includes an even number of 1-D PAM-5 symbols from family X (odd parity) and an even number from family Y (even parity). Therefore, the sum of the 1-D PAM-5 symbols is even. If the parity of the 4-D symbol is even, final

decoder 905 then outputs the symbols $\hat{a}_{k,1}$ through $\hat{a}_{k,4}$ as output symbols $\hat{a}'_{k,1}$ through $\hat{a}'_{k,4}$, respectively.

[0113] Error signals $e_{k,1}$ through $e_{k,4}$ are calculated in adders 902-1 through 902-4, respectively, by, in each of the transport channels, taking the difference between the output symbol $\hat{a}_{k,1}$ through $\hat{a}_{k,4}$ of slicers 901-1 through 901-4, respectively, and the corresponding one of the input signals $a'''_{k,1}$ through $a'''_{k,4}$, respectively:

$$e_{k,i}=a'''_{k,i}-\hat{a}_{k,i}\,,\quad i=1,\ 2,\ 3 \text{ or } 4. \tag{14}$$

The error signals $e_{k,1}$ through $e_{k,4}$ at slicers 901-1 through 901-4, respectively, are input to error analysis 904. Error analysis 904 determines which of the symbols $\hat{a}_{k,1}$ through $\hat{a}_{k,4}$ is associated with the largest error signal $e_{k,1}$ through $e_{k,4}$, respectively, and the sign of the largest error. Error analysis 904 outputs an identifier W of the symbol having the largest absolute error and the sign SGN of that error to final decoder 905. If the parity signal indicates odd parity, the erroneous symbol is most likely to be the one with the largest error, indicated by identifier W. Final decoder 905, then, adjusts the symbol having the largest error up or down the symbol alphabet by one symbol depending on the sign SGN (in this example, up if the sign is positive and down if the sign is negative) and outputs the resulting 4-D output symbol defined by $\hat{a}'_{k,1}$ through $\hat{a}'_{k,4}$.

[0114] The combination of N-D trellis code encoding with a DFE architecture requires a DFE on all four wires for each state within the trellis (*see*, e.g., Figure 3). The DFE output at a particular state is used for parallel branch decisions and branch metric computation for the branches leaving that state. The equalization in this situation uses a process called "per survivor processing", described more fully in Riccardo Raheli, Andreas Polydoros, Ching-Kai Tzou,

"Per-Survivor Processing: A General Approach to MLSE in Uncertain Environments," IEEE Trans. Commun., Vol. 43, No. 2/3/4, pp. 354-364, February/March/April 1995, incorporated by reference in its entirety, to determine the feedback symbols.

[0115] The multiple decision feedback equalizers in the trellis decoder helps to minimize the effects of error propagation. However, a DFE architecture suffers from the disadvantage that it does not utilize the sample power contained in the intersymbol interference caused by the dispersion in the channel. More particularly, information about a 1-D symbol being transmitted during a time period k is contained in future received 1-D symbols from that channel and DFE does not utilize this signal power in deciding the currently received symbol.

[0116] A third type of equalization, sequence detection, does not suffer the performance degradation of either a linear equalizer or a decision feedback equalizer. However, a typical sequence detector produces hard outputs that can severely limit the performance of an error correction code. According to the present invention, a soft-output sequence detector is provided.

[0117] As an example, Figure 10 shows a portion of a 4-D receiver utilizing sequence detection according to the present invention. The receiver shown in Figure 10 includes sequence detectors 1001-1 through 1001-4, one for each of the four transmission channels corresponding to wires 103-1 through 103-4. One skilled in the art will recognize that a receiver can include any number N of transmission channels, each of the transmission channels providing signals to one of N detectors. Some or all of the N detectors can be sequence detectors.

[0118] The output samples $\hat{a}_{k,1}$ through $\hat{a}_{k,4}$ from sequence detectors 1001-1 through 1001-4, respectively, along with the second best output samples $\hat{a}2_{k,1}$ through $\hat{a}2_{k,4}$ and the

errors Δ_1 through Δ_4 are input to decoder 1002. Decoder 1002 decides on the receiver output samples $\hat{a}'_{k,1}$ through $\hat{a}'_{k,4}$ based on, for example, the parity coding scheme or the N-D lattice coding scheme. One skilled in the art will recognize that the decision of decoder 1002 can be based on any coding scheme that can correct errors in one clock cycle. Feedback is provided to sequence detectors 1001-1 through 1001-4 so that future sequences are decided using the results of the N-D decisions.

[0119] Figure 11A shows a block diagram of a sequence detector 1100 for the PAM-5 symbol alphabet, {A}={+2, +1, 0, -1, -2}, where the intersymbol interference includes the effects of only one other symbol, i.e., the ISI length δ is one (1). In general, sequence detector 1100 can utilize any alphabet and any number of ISI symbols (A>2 and δ>1). Detector 1100 includes a branch metric generator 1101, an add-compare-select (ACS) unit 1102, traceback circuitry 1103, a last-in-first-out (LIFO) buffer 1104, and a starting point determiner 1105. Sequence detectors are discussed in U.S. Patent Application 08/974,450, cited above.

[0120] The ISI addressed by the embodiment of detector 1100 shown in Figure 11A is caused by just one previously transmitted symbol. Thus, the input sample to sequence detector 1100 is given by

$$r_{k,w} = a_{k,w} + \alpha_1 a_{k-1,w} + h_{k,w} \quad , \tag{15}$$

where $\alpha_{1,w}$ is the equalizer ISI coefficient, $h_{k,w}$ is the noise component of the output signal from the linear filter over a wire w and $a_{k,w}$ is the transmitted symbol over wire w received in time period k. The sequence detector estimates the transmitted data sequence $\{a_{k,w}\}$ from the sequence of received samples $\{r_{k,w}\}$.

[0121] The state $S_{k,w}$ of detector 1100 is defined as the past data symbol estimates. In general, a system with a symbol alphabet having A symbols and which suffers intersymbol interference from δ previous symbols will have A^{δ} possible states. Each state corresponds to a possible transition path through the δ previous symbols. For example, a system using a symbol alphabet with two symbols, {A} = {+1, -1}, and suffering ISI from two past symbols, δ=2, will have four possible sequence states of the system: symbol +1 at time k-2 and symbol +1 at time k-1; symbol +1 at time k-2 and symbol -1 at time k-1; symbol -1 at time k-2 and symbol +1 at time k-1; and symbol -1 at time k-2 and symbol -1 at time k-1. Sequence detector 1100 for the example embodiment shown in Figure 11A, a PAM-5 symbol alphabet with ISI resulting from one symbol, has five states -- +2, +1, 0, -1, and -2 -- corresponding to the five symbols in the PAM-5 alphabet {A}.

[0122] A trellis diagram depicting a state transition from time (k-1) to time k for sequence detector 1100 is shown in Figure 12. A trellis diagram provides a graphical mechanism for predicting what the signal output would be for the channel in transitions from one state in time k-1 to another state at time k. As an example, from Equation 15 above, and neglecting noise, a transition from state +2 at time k-1 to a state +1 at time k results in a signal input to sequence detector 1100 of $1+2\alpha_{w,1}$.

[0123] In sequence detector 1100 shown in Figure 11A, branch metric generator 1101 generates distance metrics corresponding to the branches illustrated in the trellis diagram of Figure 12. The distance metrics represent the difference between the input signal $r_{k,w}$ to branch metric generator 1101 and the calculated signal that is expected to be observed in each of the allowed transitions of a trellis diagram. In general, if there are A^{δ} states of sequence

detector 1100, there are $A^{\delta+1}$ branch metrics to calculate. In some embodiments, however, not all branches are allowed and therefore the number of branch metrics is reduced.

[0124] For PAM-5 signaling suffering from one ISI symbol, there are twenty five (25) distance metrics $M_{k,w}$. The twenty five (25) distance metrics generated by branch metric generator 1101 for the case where ISI is the result of one past symbol and neglecting random noise (i.e., A=5, δ=1 and $f_w(z) = 1+\alpha_{w,1}z^{-1}$) are given by:

$$
\begin{array}{ll}
M_{k,w}(0)=[r_{k,w}-(-2-2\alpha_{w,1})]^2; & M_{k,w}(13)=[r_{k,w}-(\alpha_{w,1})]^2; \\
M_{k,w}(1)=[r_{k,w}-(-2-\alpha_{w,1})]^2; & M_{k,w}(14)=[r_{k,w}-(2\alpha_{w,1})]^2; \\
M_{k,w}(2)=[r_{k,w}-(-2)]^2; & M_{k,w}(15)=[r_{k,w}-(1-2\alpha_{w,1})]^2; \\
M_{k,w}(3)=[r_{k,w}-(-2+\alpha_{w,1})]^2; & M_{k,w}(16)=[r_{k,w}-(1-\alpha_{w,1})]^2; \\
M_{k,w}(4)=[r_{k,w}-(-2+2\alpha_{w,1})]^2; & M_{k,w}(17)=[r_{k,w}-1]^2; \\
M_{k,w}(5)=[r_{k,w}-(-1-2\alpha_{w,1})]^2; & M_{k,w}(18)=[r_{k,w}-(1+\alpha_{w,1})]^2; \\
M_{k,w}(6)=[r_{k,w}-(-1-\alpha_{w,1})]^2; & M_{k,w}(19)=[r_{k,w}-(1+2\alpha_{w,1})]^2; \\
M_{k,w}(7)=[r_{k,w}-(-1)]^2; & M_{k,w}(20)=[r_{k,w}-(2-2\alpha_{w,1})]^2; \\
M_{k,w}(8)=[r_{k,w}-(-1+\alpha_{w,1})]^2; & M_{k,w}(21)=[r_{k,w}-(2-\alpha_{w,1})]^2; \\
M_{k,w}(9)=[r_{k,w}-(-1+2\alpha_{w,1})]^2; & M_{k,w}(22)=[r_{k,w}-2]^2; \\
M_{k,w}(10)=[r_{k,w}-(-2\alpha_{w,1})]^2; & M_{k,w}(23)=[r_{k,w}-(2+\alpha_{w,1})]^2; \\
M_{k,w}(11)=[r_{k,w}-(-\alpha_{w,1})]^2; & M_{k,w}(24)=[r_{k,w}-(2+2\alpha_{w,1})]^2; \\
M_{k,w}(12)=[r_{k,w}]^2. &
\end{array}
\qquad (16)
$$

[0125] Other metrics which represent the difference between the input symbol and the predicted input symbols, assuming each of the possible state transitions, may be used as distance metrics. In general, there will be a distance metric for every transition from any state S′ at time k-1 to state S at time k, $A^{\delta+1}$ distance metrics for a metrics system with A symbols and δ interfering symbols if all transitions are allowed. PAM-5 symboling is shown here only as an example and embodiments of detector 1100 can utilize other symbol alphabets and more ISI symbols.

[0126] Add-compare-select circuit 1102 in Figure 11A updates a state metric, denoted by $p_{k,w}(+2)$, $p_{k,w}(+1)$, $p_{k,w}(+0)$, $p_{k,w}(-1)$, and $p_{k,w}(-2)$ in the PAM-5 example, for each

possible state of the system at each time step k. For the PAM-5 symbol alphabet the state metrics are given by:

$$p_{k,w}(i) = \min_{j=\{-2,-1,0,1,2\}}\{p_{k-1,w}(j) + M_{k,w}((j+2)+5(i+2)\}; \quad \text{for } i=\{2,1,0,-1,-2\}. \qquad (17)$$

[0127] In general, the state metrics represent the accumulated distance metrics of past states along transition paths that minimize the accumulated distance metric. Therefore, the transition metric $p_{k,w}(S)$ for state S at time period k is the accumulated distance metric for previous states along a transition path which ends at state S at time period k, state S being one of the possible states of the system. At time k-1, the state of the system may be at any state S′ in the group of possible states of the system. Therefore, $p_{k,w}(S)$ is the minimum one of $p_{k-1,w}(S')$ plus the distance metric for transition from S′ to S. A mathematical proof that this technique results in the least detection error is given in the Appendix of U.S. Patent Application 08/974,450, cited above.

[0128] In the example of Figure 11A, the comparison results $D_{k,w}(+2)$, $D_{k,w}(+1)$, $D_{k,w}(0)$, $D_{k,w}(-1)$ and $D_{k,w}(-2)$ are stored in a memory of traceback circuit 1103 for each of the five states. The comparison results indicate the state at time period k-1 which results in the state metric $p_{k,w}(S)$ for state S at time period k. In the PAM-5, δ=1 example shown in Figure 11A, the ACS results for each of the five states are given by

$$D_{k,w}(i)=j \text{ if } p_{k,w}(i) = p_{k-1,w}(j)+M_{k,w}((j+2)+5(i+2)); \quad \text{for } i=\{2,1,0,-1,-2\};\ j=\{2,1,0,-1,-2\}. \qquad (18)$$

In general, $D_{k,w}(S)$ points toward the state S′ at time k-1 from which results the lowest state metric for arriving at state S at time k.

[0129] When it is time for decoding, traceback circuit 1103 traces back from a starting state, and fetches the data from memory in traceback circuit 1103.

[0130] In one example of a traceback, if the traceback depth is 2*TB, then it is expected that a traceback of TB is performed every TB/2 sample times and traceback circuit 1106 outputs TB/2 data symbols. With little loss of generality, TB can be an even integer such as 6, 8 or 16. A greater traceback depth will result in less error in determining the final sequence of symbols. Greater traceback depths, however, incur an implementation cost of requiring more memory in traceback circuit 1103.

[0131] During the traceback procedure, starting state determiner 1105 picks the starting state, which can be based on the state metrics $p_{k,w}(S)$. Traceback circuit 1103 follows the sequence back through the comparison results stored in memory in traceback circuit 1103. The earliest TB/2 symbols, which result in the earliest states, are written into last-in-first-out buffer 1104. The new comparison results are stored in the memory locations previously occupied by the outputted results.

[0132] Traceback circuit 1103 determines the optimum sequence of symbols based on the state metrics $p_{k,w}(S)$ stored in starting state determiner 1105. Starting state determiner 1105 initializes the traceback procedure by setting a starting sequence.

[0133] LIFO 1104 simply time-reverses the data coming out of traceback circuit 1103 because the traceback is performed from the current time to previous times.

Sequence Detection with Pre-Equalization

[0134] When the channel ISI length δ is large, or if the transmitted symbol alphabet size A is large, the above method of full sequence estimation becomes impractical at

high symbol rates. Full sequence estimations require the implementation of A^{δ} states in the detector. Accordingly, equalizer 1110 can provide pre-equalization by preprocessing the input samples $y_{k,w}$ in order to reduce the number of ISI symbols to be processed by sequence detector 1100. Equalizer 1110 can be any equalizer that reduces the number of ISI symbols. Again, for purposes of example, assume that the channel input alphabet size is A=5, i.e., {A}={+2, +1, 0, -1, -2}, and that the reduced ISI length (as seen by the sequence detector) is $\delta'=1$. As before, the technique is applicable to larger alphabets and may accommodate more than one interfering symbol in the reduced length.

[0135] The output of the equalizer 1110 with A=5 and $\delta'=1$ is, analogous to Equation 15, given by

$$r_{k,w}=a_{k,w}+\alpha_{w,1}*a_{k-1,w}+h_{k,w}, \quad (19)$$

where $\alpha_{w,1}$ is the equalized ISI coefficient and h_k is the noise component of the output of the linear equalizer 1110. The transfer function (in Z-transform notation, *see* A. V. OPPENHEIM AND R.W. SCHAFER, DISCRETE-TIME SIGNAL PROCESSING, (1989)) for equalizer 1110 is then given by

$$E_w(z)=(1+\alpha_{w,1}z^{-1})/f_w(z). \quad (20)$$

The coefficient $\alpha_{w,1}$ is chosen to minimize the noise variance at the output of the equalizer. Equalizer 1110, therefore, is a reduced sequence equalizer because it reduces the ISI length from δ to δ'. The reduced ISI length δ' is one (1) in this example.

[0136] In one embodiment, reduced sequence equalizer 1110 is implemented adaptively. One architecture used for

adaptive implementation is shown in Figure 11B. In this embodiment, equalizer 1110 includes a linear equalizer 1120 implementing a transfer function $C(Z) = 1/f(Z) = C_0 + C_1Z^{-1} + \ldots + C_\delta Z^{-\delta}$, adaptively followed by a filter 1121 implementing the function $(1+\alpha_{w,1}Z^{-1} + \ldots +\alpha_{w,\delta'}Z^{-\delta'})$. By implementing both equalizer 1120 and filter 1121 adaptively, optimal performance can be achieved for any cable length. Linear equalizer 1120 can be adaptively implemented by using the least mean squares (LMS) algorithm (*see* E.A. LEE AND D.G. MESSERCHMITT, DIGITAL COMMUNICATIONS (1988)) and a finite impulse response filter as shown in Figure 6.

[0137] The coefficients $\alpha_{w,1}$ through $\alpha_{w,\delta'}$ can be chosen adaptively in the sequence detector by observing the frequency response of linear equalizer 1120. From linear equalizer 1120, the channel frequency response is deduced and $\alpha_{w,1}$ through $\alpha_{w,\delta'}$ can be selected from a look-up table. In one embodiment with δ'=1, two possible values of $\alpha_{w,1}$ (0 and 1/2) are used. One of the two possible coefficients is chosen for $\alpha_{w,1}$ by observing the two largest equalizer coefficients of linear equalizer 1120, C_0 and C_1. For example, in one embodiment, if $C_1/C_0 < -0.5$ then $\alpha_{w,1} = 0.5$, otherwise $\alpha_{w,1} = 0$.

[0138] The benefits of combining linear equalization with sequence detection include (a) reduced complexity in the sequence detector, especially for large ISI lengths, and (b) reduced noise enhancement in the linear equalization.

[0139] In the example described in connection with Figure 11B with δ'=1, the number of states in the sequence estimator is reduced from 5^δ to 5. The reduced state sequence estimator can be implemented using the Viterbi algorithm.

Sequence Detection with Pre-equalization and Decision Feedback

[0140] Figure 13 shows a receiver 1350 that includes an embodiment of a sequence detector 1300 having decision-feedback. In this embodiment, equalizer 1301, which as before can be any equalizer structure, pre-equalizes transmission channel 104-w to a pre-determined ISI polynomial $G_w(z)$ of length $\eta \le \delta$, where δ is the ISI length of the frequency response $f_w(z)$ of transmission channel 104-w. In one example, η is 2 and the ISI polynomial $G_w(z)$ is given by

$$G_w(z) = 1 + \alpha_{w,1}z^{-1} + \alpha_{w,2}z^{-2}. \qquad (21)$$

The transfer function of equalizer 1301 is given by

$$E_w(z) = G_w(z)/f_w(z). \qquad (22)$$

[0141] Sequence detector 1300 includes branch metric generator 1302, add compare select (unit) 1303, traceback (circuit) 1304, LIFO 1305 and starting point determiner 1306. In general, the detection technique implemented in sequence estimator 1300 may be used for any combination of δ and η such that $\eta \le \delta$. Although the technique may be implemented with any sized alphabet, the example shown in Figure 13 is for the PAM-5, (A=5), alphabet. The coefficients, $\alpha_{w,1}$ and $\alpha_{w,2}$, can be chosen adaptively to optimize performance of the receiver 1350.

[0142] To perform the sequence estimation using the Viterbi algorithm, as outlined above, sequence detector 1300 is still implemented with twenty five states (if the data symbols are PAM-5 values). The branch metric computations from a trellis diagram now accounts for the intersymbol

interference due to the symbols transmitted two sample times before as well as the symbol transmitted during the last period. In one embodiment, the branch metrics computed by branch metric generator 1302 with decision feedback are given by:

$$M_{k,w}(0)=[r_{k,w}-\alpha_{w,2}D_{k-1}(-2)-(-2-2\alpha_{w,1})]^2;$$
$$M_{k,w}(1)=[r_{k,w}-\alpha_{w,2}D_{k-1}(-1)-(-2-\alpha_{w,1})]^2;$$
$$M_{k,w}(2)=[r_{k,w}-\alpha_{w,2}D_{k-1}(0)-(-2)]^2;$$
$$M_{k,w}(3)=[r_{k,w}-\alpha_{w,2}D_{k-1}(1)-(-2+\alpha_{w,1})]^2;$$
$$M_{k,w}(4)=[r_{k,w}-\alpha_{w,2}D_{k-1}(2)-(-2+2\alpha_{w,1})]^2;$$
$$M_{k,w}(5)=[r_{k,w}-\alpha_{w,2}D_{k-1}(-2)-(-1-2\alpha_{w,1})]^2;$$
$$M_{k,w}(6)=[r_{k,w}-\alpha_{w,2}D_{k-1}(-1)-(-1-\alpha_{w,1})]^2;$$
$$M_{k,w}(7)=[r_{k,w}-\alpha_{w,2}D_{k-1}(0)-(-1)]^2;$$
$$M_{k,w}(8)=[r_{k,w}-\alpha_{w,2}D_{k-1}(1)-(-1+\alpha_{w,1})]^2;$$
$$M_{k,w}(9)=[r_{k,w}-\alpha_{w,2}D_{k-1}(2)-(-1+2\alpha_{w,1})]^2;$$
$$M_{k,w}(10)=[r_{k,w}-\alpha_{w,2}D_{k-1}(-2)-(-2\alpha_{w,1})]^2;$$
$$M_{k,w}(11)=[r_{k,w}-\alpha_{w,2}D_{k-1}(-1)-(-\alpha_{w,1})]^2;$$
$$M_{k,w}(12)=[r_{k,w}-\alpha_{w,2}D_{k-1}(0)]^2;$$
$$M_{k,w}(13)=[r_{k,w}-\alpha_{w,2}D_{k-1}(1)-\alpha_{w,1}]^2;$$
$$M_{k,w}(14)=[r_{k,w}-\alpha_{w,2}D_{k-1}(2)-2\alpha_{w,1}]^2;$$
$$M_{k,w}(15)=[r_{k,w}-\alpha_{w,2}D_{k-1}(-2)-(1-2\alpha_{w,1})]^2;$$
$$M_{k,w}(16)=[r_{k,w}-\alpha_{w,2}D_{k-1}(-1)-(1-\alpha_{w,1})]^2;$$
$$M_{k,w}(17)=[r_{k,w}-\alpha_{w,2}D_{k-1}(0)-1]^2;$$
$$M_{k,w}(18)=[r_{k,w}-\alpha_{w,2}D_{k-1}(1)-(1+\alpha_{w,1})]^2;$$
$$M_{k,w}(19)=[r_{k,w}-\alpha_{w,2}D_{k-1}(2)-(1+2\alpha_{w,1})]^2;$$
$$M_{k,w}(20)=[r_{k,w}-\alpha_{w,2}D_{k-1}(-2)-(2-2\alpha_{w,1})]^2;$$
$$M_{k,w}(21)=[r_{k,w}-\alpha_{w,2}D_{k-1}(-1)-(2-\alpha_{w,1})]^2;$$
$$M_{k,w}(22)=[r_{k,w}-\alpha_{w,2}D_{k-1}(0)-2]^2;$$
$$M_{k,w}(23)=[r_{k,w}-\alpha_{w,2}D_{k-1}(1)-(2+\alpha_{w,1})]^2;$$
$$M_{k,w}(24)=[r_{k,w}-\alpha_{w,2}D_{k-1}(2)-(2+2\alpha_{w,1})]^2. \qquad (23)$$

[0143] The ISI due to the transmitted symbol at time (k-2) is removed from the received sample $r_{k,w}$ before the branch metric of the sequence detector that accounts for the ISI due to the transmitted symbol at time (k-1) is computed. After this calculation, the previously described calculation

is performed to remove the ISI due to the (k-1)th transmitted symbol.

[0144] Add compare select 1303 then computes the state metrics $p_{k,w}(j)$ and the comparison results $D_{k,w}(j)$ as described above in Equations 17 and 18, respectively. Traceback 1304 accepts a starting point from starting point determiner 1306, as described above for the corresponding components of sequence detector 1100 in Figure 11A, and outputs a set of decided-on symbols to LIFO 1305. LIFO 1305, then, outputs the resulting symbols in reverse chronological order from that received.

Sequence Detection in Combination with Error Correction

[0145] Although achieving excellent noise margins, the combination of sequence detection with error correction codes is problematic. Sequence detectors usually produce hard decisions (i.e., decisions that do not contain information on reliability), which become input signals to error correcting codes. Most error correcting techniques, such as the parity coding or 4-D lattice coding proposed for Gigabit Ethernet, rely on soft decisions (i.e., decisions that contain reliability information) to achieve full performance. *See*, e.g., S. LIN AND D. J. COSTELLO, JR., ERROR CONTROL CODING: FUNDAMENTALS AND APPLICATIONS (1983). Thus, a typical sequence detector in combination with a decoder using an error correcting code does not achieve the improved SNR margins of both the sequence detector and the error correcting code unless the decoder is provided with a soft decision from the sequence detector.

[0146] Figure 14 shows a sequence detector 1400-w according to the present invention. Sequence detector 1400-w is coupled to receive signal $r_{k,w}$ from wire w. Sequence detector 1400-w includes branch metric generator 1402-w, add compare select (unit) 1403-w, and starting point determiner

1406-w. Add compare select 1402-w and starting point determiner 1406-w are each coupled to traceback 1404.

[0147] Traceback 1404 is coupled to each wire w so that the traceback can be accomplished on N-D symbols rather than performing a separate traceback on N 1-D symbols, where N is the number of wires. Traceback 1404 is also coupled with LIFO 1405. LIFO 1405 outputs the final symbol stream $\hat{a}'_{k,w}$ for each wire w.

[0148] A sequence detector according to the present invention can accommodate any symbol alphabet and any number of ISI symbols. For example, sequence detector 1400 shown in Figure 14 is for a PAM-5 alphabet where input signals suffer the influence of one past ISI symbol (i.e., the ISI length is 1). As occurs with equalizer 1110 in Figure 11A (*see* Equation 20), equalizer 1401-w executes a transfer function $E(Z) = (1+\alpha_{w,1}Z^{-1})/f_w(Z)$ and therefore sequence detector 1400-w detects a signal that includes ISI from one past symbol. Branch metric generator 1402-w therefore generates the branch metrics as described in Equation 16.

[0149] Add compare select 1403-w computes the state metrics $p_{k,w}(i)$ $(i=-2,-1,0,1,2)$ according to Equation 17 and the ACS results $D_{k,w}(i)$ according to Equation 18. Additionally, ACS 1403-w computes a second best state metric $p2_{k,w}(i)$, a second ACS result $D2_{k,w}(i)$, and a difference result $\Delta_{k,w}(i)$. The second best state metric can be computed according to the equation

$$p2_{k,w}(i) = \text{second minimum}\,(pk\text{-}1,w(j) + Mk,w(5(i+2)+(j+2)));\quad j=\{-2, -1, 0, 1, 2\}, \tag{24}$$

where i={-2, -1, 0, 1 2}. The second ACS result can be computed according to

$$D2_{k,w}(i) = j \text{ if } p2_{k,w}(i) = p_{k-1,w}(j) + M_{k,w}(5(i+2)+(j+2)), \quad (25)$$

where i={-2,-1,0,1,2} and j=(-2,-1,0,1,2}. Finally, the difference result can be computed according to

$$\Delta_{k,w}(i) = p2_{k,w}(i) - p_{k,w}(i). \quad (26)$$

Traceback 1404 receives the ACS results, the second ACS results, and the difference results from each state on each wire as well as a starting point signal from starting point determiner 1406-w for each wire w.

[0150] Figure 15 shows an embodiment of 4-D traceback 1404 for use with a series of sequence detectors 1400-w. For an N-wire configuration, traceback 1404 is generally coupled into N sequence detectors similar to that shown in Figure 14, one for each wire w. In Figure 15, traceback 1404 includes a parity code decoder 1504 although other coding schemes can be used, provided that the coding scheme can correct symbols within one clock cycle.

[0151] For exemplary purposes, Figure 15 shows a four-wire parity-code decoder. Traceback 1404 includes read modules 1501-1 through 1501-4, one for each of the four wires. Each read module 1501-1 through 1501-4 receives parameters $\Gamma_{k,w}$ from add compare select 1403-w and starting point determiner 1406-w (Figure 14). Parameters $\Gamma_{k,w}$ includes the add compare results $D_{k,w}(i)$, $D2_{k,w}(i)$ and $\Delta_{k,w}(i)$, where i=(-2, -1, 0, 1, 2) for PAM-5 signaling, as well as a starting point $SP_{k,w}$ for each wire w. As was discussed before, starting point determiner 1406-w chooses a starting point $SP_{k,w}$, which can be based on the state metrics $p_{k,w}(i)$, for traceback 1404.

[0152] Traceback 1404 in Figure 15 traces back to arrive at the best sequence $\hat{a}_{k,w}$ for wire w. For each clock cycle k, read module 1501-w arrives at a best symbol $\hat{a}_{k,w}$, a second

best symbol $\hat{a}2_{k,w}$ and an associated reliability $\varepsilon_{k,w}$ for each wire w.

[0153] As described above, the 4-D parity coding scheme only transmits 4-D symbols having even parity. A single error in one 1-D symbol will cause the parity of the 4-D symbol to become odd. Similar to "4-D slicing", traceback 1404 recognizes the parity error and makes corrections to the four 1-D symbols from read modules 1501-1 through 1501-4 for clock cycle k based on the reliability of each of the 1-D symbols.

[0154] Hagenauer has shown that, within a Viterbi decoder, the reliability of decision symbol paths merging at each state grows with the difference between the state metrics between the two paths. *See* J. Hagenauer and P. Hocher, "A Viterbi Algorithm with Soft-Decision Outputs and its Applications," Proc. GLOBECOM '89, 1680-1686, November 1989. Similar to Hageauer's Soft Output Viterbi Algorithm (SOVA), each of read modules 1501-1 through 1501-4 outputs the difference metric, i.e., the reliability measure, between the best two paths entering each state, $\varepsilon_{k,1}$ through $\varepsilon_{k,4}$, respectively. SOVA uses these metrics over a range of sample times within the trellis to output soft decisions for every symbol.

[0155] However, 4-D parity traceback 1404 can recognize when a single error occurs. Therefore, only the difference metrics at the time of the error are required to correct errors. Because the actual decoding is accomplished during traceback in each of sequence detectors 140-1 through 140-4, traceback 1404 recognizes an error in its channel and corrects the error in its own sequence path.

[0156] For each clock cycle k, traceback 1404 retrieves each of the first choice 1-D symbols $\hat{a}_{k,1}$ through $\hat{a}_{k,4}$, the second choice 1-D symbols $\hat{a}2_{k,1}$ through $\hat{a}2_{k,4}$, and the

reliability measures $\varepsilon_{k,1}$ through $\varepsilon_{k,4}$, and determines the finally decided four 1-D symbols $\hat{a}'_{k,1}$ through $\hat{a}'_{k,4}$.

[0157] For each clock cycle k, parity check 1502 receives the first choice symbols $\hat{a}_{k,1}$ through $\hat{a}_{k,4}$ from read modules 1501-1 through 1501-4, respectively, determines the parity of the resulting 4-D symbol and outputs a parity signal indicating the parity of the 4-D symbol. Error analysis 1503 receives reliability measures $\varepsilon_{k,1}$ through $\varepsilon_{k,4}$ from read modules 1501-1 through 1501-4, respectively, determines which of the sequence detectors w has the least reliability, thereby indicating which result is most likely to be incorrect, and outputs a wire signal W indicating which symbol is most likely to be incorrect. Decoder 1504 receives the first choice symbols $\hat{a}_{k,1}$ through $\hat{a}_{k,4}$, the second choice symbols $\hat{a}2_{k,1}$ through $\hat{a}2_{k,4}$, the parity signal from parity check 1502, and the wire signal from error analysis 1503.

[0158] If the parity signal indicates that the parity of the first choice 4-D symbol is even, then the first choice symbols $\hat{a}_{k,1}$ through $\hat{a}_{k,4}$ are output as the finally decided symbols $\hat{a}'_{k,1}$ through $\hat{a}'_{k,4}$. However, an odd parity indicates an error in one of the first choice symbols $\hat{a}_{k,1}$ through $\hat{a}_{k,4}$. If the parity signal indicates an odd parity, the symbol indicated by the wire signal W is replaced by a corresponding one of second choice symbols $\hat{a}2_{k,1}$ through $\hat{a}2_{k,4}$ and the resulting 4-D symbol is output as finally decided symbols $\hat{a}'_{k,1}$ through $\hat{a}'_{k,4}$.

[0159] Additionally, decoder 1504 informs read modules 1501-1 through 1501-4 of the finally decided symbols $\hat{a}'_{k,1}$ through $\hat{a}'_{k,4}$ by indicating, for each of read modules 1501-1 through 1501-4, which of the first choice symbols or the second choice symbols was output as the finally decided symbols for clock cycle k. Read modules 1501-1 through 1501-4 can then traceback accordingly and output the first choice symbols, second choice symbols, and reliability

measures for clock cycle k-1 (i.e., $\hat{a}_{k-1,1}$ through $\hat{a}_{k-1,4}$, $\hat{a}2_{k-1,1}$ through $\hat{a}2_{k-1,4}$, and $\varepsilon_{k-1,1}$ through $\varepsilon_{k-1,4}$, respectively).

[0160] In other words, read module 1501-w outputs the best possible step back $\hat{a}_{k,w}$, the second best step back $\hat{a}2_{k,w}$, and the reliability measure $\varepsilon_{k,w}$ for each wire for clock cycle k. Parity check 1502 performs a parity check on the best possible step back $\hat{a}_{k,w}$. Error analysis 1503 determines the wire most likely to be incorrect. If the parity passes, decoder 1504 outputs the best possible step back $\hat{a}_{k,w}$ for clock cycle k. If parity fails, then decoder 1504 replaces one of the best possible symbols with the associated second best symbol, based on which wire is most likely to be incorrect, and outputs the resulting 4-D symbol. The choice of best symbol or second best symbol for each wire w is communicated back to read module 1501-w so that read module 1501-w can use the appropriate symbol to step back to clock cycle k-1. The replacement choice then affects only one read module. Therefore, the next set of symbols will be affected in one read module only.

[0161] As shown in Figure 14, the output symbols $\hat{a}'_{k,w}$ are output to LIFO 1405, which time reverses the order of the symbol stream and outputs the resulting symbol stream $\hat{a}'_{k,w}$.

[0162] One skilled in the art will recognize that in general a sequence detector according to this invention can accommodate any symbol alphabet and the effects of any number of ISI symbols. Additionally, the sequence detector may include decision feedback as shown in Figure 13.

Reduced Complexity Sequence Detection Using State Reduction

[0163] If the alphabet size is large or if the ISI length at the sequence detector is large then sequence detector 1400-w shown in Figure 14 and traceback 1404 using parity coding shown in Figure 15 become impractical at high symbol rates. The number of states required in read modules 1501-1

through 1501-4 is A^{η} on each wire, where A represents the number of symbols in alphabet {A} and η represents the ISI symbol length at the sequence detector (i.e., at the input terminal of sequence detector 1400 in Figure 14). With a PAM-5 alphabet and η=2 ISI symbols, each of read modules 1501-1 through 1501-4 requires twenty five (25) states in order to perform sequence detection. A decoder utilizing a large number of states is expensive, difficult to implement, and consumes a lot of power.

[0164] Figure 16 shows a sequence detector 1600-w having reduced complexity sequence detection. In particular, sequence detector 1600-w includes branch metric generator 1602-w, add-compare-select (unit) 1603-w, and starting point determiner 1605-w. Traceback 1604 and LIFO 1606 are coupled into sequence detector 1600-2 as well as similar detectors coupled to the remaining N wires. As an example, sequence detector 1600-w is shown for the PAM-5 alphabet, although embodiments of sequence detector 1600-w can utilize other symbol alphabets as well.

[0165] As mentioned above, the PAM-5 symbol alphabet can be segregated into two families, an odd family X having the PAM-5 symbols {-1, +1} and an even family Y having the PAM-5 symbols {-2, 0, +2}. A detector state can now be defined as the previous η families X or Y, as opposed to the previous η PAM-5 symbols {-2, -1, 0, +1, +2}. Therefore, the number of states required for the PAM-5 symbol alphabet with η=2 ISI symbols is reduced from 25 states to 4 states on each transmission channel w=1 through L. For a four wire system (4-D decoding, for example), there are a total of sixteen (16) states instead of one hundred (100) states.

[0166] Reduced state detection can be accomplished when the minimum squared distance between any two parallel branches of a state transition $S \rightarrow S'$ exceeds the minimum squared distance between any two paths in the trellis. A parallel

branch refers to transitions S→S′ between individual states through different symbols. For example, a state S = X can transition to a state S′ = X through receipt of either a -1 symbol or a +1 symbol.

[0167] The actual squared minimum distance between sequences does not decrease and thus the performance suffers little or no degradation from that of full state sequence detection. Gigabit Ethernet using a PAM-5 symbol set suffering ISI from two (2) symbols meets this criteria.

[0168] Figure 17 shows a trellis diagram displaying the reduced state transitions and all parallel PAM-5 transitions. In Figure 17, a state is represented by two consecutive symbols (reflecting an ISI length of 2). A transition from state S at time k-1 to state S′ at time k has two or three parallel branches, depending upon whether the new symbol entering at state S′ is in family X or family Y. For example, a transition from state XX at time k-1 to state XX at state k has two branches because the incoming symbol must be either a -1 or +1 PAM-5 symbol. A transition from state XX at time k-1 to a state XY at time k, however, indicates that either a -2, 0 or +2 symbol is received and therefore there are three branches.

[0169] In Figure 16, equalizer 1601-w can execute the transfer function

$$E(Z) = (1+\alpha_1 Z^{-1}+\alpha_2 Z^{-2})/f(Z). \qquad (27)$$

Therefore, the expected signal input to decoder system 1600 is

$$r_{k,w} = a_{k,w}+\alpha_1 a_{k-1,w}+\alpha_2 a_{k-2,w}. \qquad (28)$$

[0170] Branch metric generator 1602-w computes the branch metrics for the state transitions displayed in Figure 17. In the reduced state trellis shown in Figure 17, branch metric 1602-w first decides on which of the two or three branches of the transition from state S to state S′ to assign to the state transition $S \rightarrow S'$. The decision of which branch performs the $S \rightarrow S'$ transition is based on calculating a distance metric for each branch and choosing the branch having the lowest metric. For example, in the transition from state S=XX to state S′=XX branch metric generator first determines which of the two branches (-1 or +1) are traversed before calculating the metric. The decision of which branch is traversed is stored in branch decision $B_{k,w}(S \rightarrow S')$. The branch decision and the metric are both communicated to ACS 1603-w.

[0171] For each state S at k-1, symbols $\hat{a}_{k-1}(S)$ and $\hat{a}_{k-2}(S)$ are known based upon feedback from ACS 1603-w to branch metric generator 1602-w, as shown in Figure 16. For each state transition $S \rightarrow S'$, branch metric generator takes the difference between the input signal and the ISI portion of Equation 28 to obtain a difference

$$\sigma(S \rightarrow S') = r_{k,w} - \alpha_1 \hat{a}_{k-1,w}(S) - \alpha_2 \hat{a}_{k-2,w}(S). \qquad (29)$$

The difference σ is then compared with the symbols for each possible branch of the state transition S to S′. The symbol chosen $a'_{k,w}(S \rightarrow S')$ for the branch is then assigned to the transition from state S to state S′.

[0172] With reference to Figure 18, the branch metrics can be computed for each transition S to S′ according to the equation:

$$M_{k,w}(S \rightarrow S') = [r_{k,w} - a'_{k,w}(S \rightarrow S') - \alpha_1 \hat{a}_{k-1,w}(S) - \alpha_2 \hat{a}_{k-2,w}(S)]^2, \quad (30)$$

where the valid transitions ($S \rightarrow S'$) in Figure 18 are $S \rightarrow S'$ = XXX, XXY, YXX, YXY, XYX, XYY, YYX and YYY. The notation "ABC" used above and in Figure 18, where each of "A", "B", and "C" is X or Y, indicates an $S \rightarrow S'$ transition from S=AB to S'=BC.

[0173] Add-compare-select 1603-w receives the branch metrics $M_{k,w}((S \rightarrow S')$ and branch decisions $B_{k,w}(S \rightarrow S')$ from branch metric generator 1602-w and calculates the state metrics $P_{k,w}(S')$ according to the equation

$$P_{k,w}(S') = \min_{j=\{S\}}(P_{k-1,w}(j) + M_{k,w}(S \rightarrow S')), \quad (31)$$

where j is equal to each S such that $S \rightarrow S'$ is allowable.

[0174] Add-compare-select 1603-w also determines the ACS results for each of the four final states S' and communicates those results to traceback circuit 1604. The ACS result $D_{k,w}(S')$ is the path resulting in state S' having the lowest metric $M(S \rightarrow S')$. Because there are only two allowed paths that result in state S' (*see* Figures 17 and 18), the second best choice is automatically the path that is contraindicated by $D_{k,w}(S')$ and need not be separately stored.

[0175] The error $\Delta_{k,w}(S')$ is the difference in metrics between the two paths resulting in state S':

$$\Delta_{k,w}(S') = \left| M(S_1 \rightarrow S') - M(S_2 \rightarrow S') \right|, \quad (32)$$

where S_1 is one of the two initial states that transition to the final state S' and S_2 is the other of the two initial

states that transition to the final state S'. For example, from Figure 18, if S'=YX, then S_1 and S_2 are the states XY and YY, respectively.

[0176] Finally, the branch decisions of branch metric generator 1602-w are also communicated to traceback circuit 1604. Branch decision $B1_{k,w}(S')$ corresponds to the decision on the path $S \rightarrow S'$ indicated by $D_{k,w}(S')$. Branch decision $B2_{k,w}(S')$ corresponds to the decision on the path $S \rightarrow S'$ contraindicated by $D_{k,w}(S')$. Therefore, in the soft-decision process, if the traceback circuit 1604 is altered to the second most likely path, the parallel path decision for the second most likely path is available to traceback circuit 1604.

[0177] Traceback circuit 1604 can be the same as traceback 1404 of Figure 15 except that read modules 1501-1 through 1501-4 receive a different set of parameters $\Gamma_{k,w}$. In traceback circuit 1604, $\Gamma_{k,w}$ includes the ACS parameters $D_{k,w}(S_i)$, $\Delta_{k,w}(S_i)$, $B1_{k,w}(S_i)$, $B2_{k,w}(S_i)$, where S_i = (XX, XY, YX, YY), and the starting point $SP_{k,w}$ from starting point determiner 1605-w. As described above, read modules 1501-1 through 1501-4 store the ACS results in memory and, starting from starting point $SP_{k,w}$ determined by starting point determiner 1605, determine the most likely symbol $\hat{a}_{k,w}$ and the second most likely symbol $\hat{a}2_{k,w}$ for clock cycle k. A reliability measure $\varepsilon_{k,w}$ indicating the difference in distance metrics between the most likely symbol and the second most likely symbol is also determined.

[0178] As is shown in Figure 15, read modules 1501-1 through 1501-4 output the parameters $\hat{a}_{k,w}$ to parity check 1502 and decoder 1504, parameters $\hat{a}2_{k,w}$ to decoder 1504, and parameters $\varepsilon_{k,w}$ to error analysis 1503. As was previously discussed, decoder 1504 outputs the best symbols $\hat{a}'_{k,w}$ based on the parity check and informs read modules 1501-1 through 1501-4 of the choice. Read modules 1501-1 through 1501-4

then proceed to determine most likely and second most likely symbols for clock cycle k-1 until the traceback is complete.

[0179] In sequence detector 1600-w of Figure 16, the minimum squared distance between parallel branches of any transition S→S′ should be at least twice as large as the minimum squared distance of the overall decoder. Because there is no protection between parallel branch decisions S→S″ of sequence detector 1600-w, this requirement on minimum squared distances should hold true in order to prevent the overall minimum distance from decreasing because of reduced state detection.

[0180] As was previously discussed, the 4-D parity code provides 3 dB of coding gain, which doubles the squared minimum distance between possible paths. However, this code provides no protection between parallel branches of transitions along one wire. Therefore, this overall requirement on minimum distances prevents the minimum squared distance from decreasing because of reduced state detection. This requirement holds for Gigabit Ethernet standards, as described above. Although the requirement should hold in general, it may not be held for all embodiments of the invention.

Simplified DFE

[0181] Reduced sequence detection can also be accomplished utilizing decision feedback equalization (DFE). However, a simplified DFE provides a simple and easy implementation for accomplishing the equalization process in the allotted time (8 ns in the example of gigabit transmission over four wires).

[0182] Figure 19 shows an example of a simplified decision feedback equalizer 1900. The simplified decision feedback equalizer includes a pre-equalizer section 1901, an adder 1902, a slicer 1903, and a feedback section 1905.

Pre-equalizer section 1901 can be any equalizer structure that reduces the ISI length to L symbols. Pre-equalizer section 1901, therefore, executes the transfer function

$$T(Z) = \frac{1 + \alpha_1 Z^{-1} + \cdots + \alpha_L Z^{-L}}{f(Z)}, \quad (33)$$

where α_1 through α_L are the multiplier coefficients of pre-equalizer section 1901 and f(Z) is the response of the input channel (*see* Equation 2 for the response of a transfer channel).

[0183] The output signal a'_k from pre-equalizer (or feedforward) section 1901 is input to adder 1902. Adder 1902 subtracts the signal a''_k from selector 1906 from the output signal a'_k from feedforward section 1901. The resulting signal $a'''_k = a'_k - a''_k$ is input to slicer 1903. Slicer 1903 outputs a symbol $\hat{a}_k$ that is closest to the input signal a'''_k. The feedback section 1905 (see also feedback section 811 of Figure 8) of decision feedback equalizer 1900 comprises delays 1904-1 through 1904-L. Selector 1906 receives each of L past symbols $\hat{a}_{k-1}$ through $\hat{a}_{k-L}$ and uses these symbols to access a lookup table. The lookup table holds values ξ_1 through ξ_Q. The output signal a''_k of selector 1906 then is that one of ξ_1 through ξ_Q that corresponds to the combination of inputs $\hat{a}_{k-1}$ through $\hat{a}_{k-L}$. The time required to look the results up in a look-up table is much less than the time required to perform the L multiplications and L additions required of the feedback section shown, for example, as feedback section 811 of Figure 8.

[0184] In some embodiments, selector 1906 receives the look-up values ξ_1 through ξ_Q as input signals. In some embodiments, the look-up values ξ_1 through ξ_Q are preset.

The look-up values ξ_1 through ξ_Q can also be adaptively chosen to optimize performance of the receiver of which decision feedback equalizer 1900 is a part. In most embodiments, $Q = A^L$ where A is the size of the symbol alphabet.

[0185] As an example, in a system using the PAM-5 alphabet where L=2, and Q is 25 there are twenty-five (25) lookup values (i.e., Q=25). Because the intersymbol interference in the input signal to adder 1902 is the result of two (2) ISI symbols,

$$a'_k = a_k + \alpha a_{k-1} + \beta a_{k-2} + n_k, \tag{34}$$

where α and β are the interference parameters and n_k is random noise. The twenty-five values for the look-up table, therefore, are given by:

$$\begin{array}{llll}
a''_k = \xi_1 = 2\alpha+2\beta & \text{if } (\hat{a}_{k-1} = 2) & \text{and} & (\hat{a}_{k-2} = 2); \\
a''_k = \xi_2 = 2\alpha+\beta & \text{if } (\hat{a}_{k-1} = 2) & \text{and} & (\hat{a}_{k-2} = 1); \\
a''_k = \xi_3 = 2\alpha & \text{if } (\hat{a}_{k-1} = 2) & \text{and} & (\hat{a}_{k-2} = 0); \\
a''_k = \xi_4 = 2\alpha-\beta & \text{if } (\hat{a}_{k-1} = 2) & \text{and} & (\hat{a}_{k-2} = -1); \\
a''_k = \xi_5 = 2\alpha-2\beta & \text{if } (\hat{a}_{k-1} = 2) & \text{and} & (\hat{a}_{k-2} = -2); \\
a''_k = \xi_6 = \alpha+2\beta & \text{if } (\hat{a}_{k-1} = 1) & \text{and} & (\hat{a}_{k-2} = 2); \\
a''_k = \xi_7 = \alpha+\beta & \text{if } (\hat{a}_{k-1} = 1) & \text{and} & (\hat{a}_{k-2} = 1); \\
a''_k = \xi_8 = \alpha & \text{if } (\hat{a}_{k-1} = 1) & \text{and} & (\hat{a}_{k-2} = 0); \\
a''_k = \xi_9 = \alpha-\beta & \text{if } (\hat{a}_{k-1} = 1) & \text{and} & (\hat{a}_{k-2} = -1); \\
a''_k = \xi_{10} = \alpha-2\beta & \text{if } (\hat{a}_{k-1} = 1) & \text{and} & (\hat{a}_{k-2} = -2); \\
a''_k = \xi_{11} = 2\beta & \text{if } (\hat{a}_{k-1} = 0) & \text{and} & (\hat{a}_{k-2} = 2); \\
a''_k = \xi_{12} = \beta & \text{if } (\hat{a}_{k-1} = 0) & \text{and} & (\hat{a}_{k-2} = 1); \\
a''_k = \xi_{13} = 0 & \text{if } (\hat{a}_{k-1} = 0) & \text{and} & (\hat{a}_{k-2} = 0); \\
a''_k = \xi_{14} = -\beta & \text{if } (\hat{a}_{k-1} = 0) & \text{and} & (\hat{a}_{k-2} = -1); \\
a''_k = \xi_{15} = -2\beta & \text{if } (\hat{a}_{k-1} = 0) & \text{and} & (\hat{a}_{k-2} = -2); \\
a''_k = \xi_{16} = -\alpha+2\beta & \text{if } (\hat{a}_{k-1} = -1) & \text{and} & (\hat{a}_{k-2} = 2); \\
a''_k = \xi_{17} = -\alpha+\beta & \text{if } (\hat{a}_{k-1} = -1) & \text{and} & (\hat{a}_{k-2} = 1); \\
a''_k = \xi_{18} = -\alpha & \text{if } (\hat{a}_{k-1} = -1) & \text{and} & (\hat{a}_{k-2} = 0); \\
a''_k = \xi_{19} = -\alpha-\beta & \text{if } (\hat{a}_{k-1} = -1) & \text{and} & (\hat{a}_{k-2} = -1); \\
a''_k = \xi_{20} = -\alpha-2\beta & \text{if } (\hat{a}_{k-1} = -1) & \text{and} & (\hat{a}_{k-2} = -2); \\
a''_k = \xi_{21} = -2\alpha+2\beta & \text{if } (\hat{a}_{k-1} = -2) & \text{and} & (\hat{a}_{k-2} = 2); \\
a''_k = \xi_{22} = -2\alpha+\beta & \text{if } (\hat{a}_{k-1} = -2) & \text{and} & (\hat{a}_{k-2} = 1);
\end{array}$$

$$a''_k = \xi_{23} = -2\alpha \quad \text{if } (\hat{a}_{k-1} = -2) \text{ and } (\hat{a}_{k-2} = 0);$$
$$a''_k = \xi_{24} = -2\alpha-\beta \quad \text{if } (\hat{a}_{k-1} = -2) \text{ and } (\hat{a}_{k-2} = -1);$$
$$a''_k = \xi_{25} = -2\alpha-2\beta \quad \text{if } (\hat{a}_{k-1} = -2) \text{ and } (\hat{a}_{k-2} = -2). \quad (35)$$

[0186] The parameters α and β can be adaptively chosen and the table updated periodically by calculating the look-up values ξ_1 through ξ_Q and inputting them into selector 1906. When decision feedback equalizer 1900 is utilized, for example, as equalizer 505-j in receiver 501-j of Figure 5B, look-up values ξ_1 through ξ_Q can be calculated by coefficient update 506-j and read into equalizer 505-j.

[0187] Figure 20A shows a sequence detector 2000. Sequence detector 2000 can be any sequence detector system having feedback, including sequence detector 1300 (Figure 13) and sequence detector 1600 (Figure 16). As before, sequence detector 2000 utilizes any symbol alphabet and includes trellis decoding for any number of past ISI symbols.

[0188] Pre-equalizer section 2001 of sequence detector 2000 receives signal $y_{k,w}$, and executes a transfer function, such as that shown in Equation 33, that removes the ISI influence from all but L past symbols. Feedback section 2003 outputs a signal $a''_{k,w}$ that removes the influence from an additional M past symbols based on the inputs from the ACS. Sequence detector 2000 therefore utilizes states describing the past L-M ISI symbols. Figure 12, for example, illustrates a trellis diagram for L-M = 1. Figures 17 and 18, for example, illustrate a trellis diagram for a reduced state sequence detector with L-M = 2.

[0189] As occurs in branch metric generators 1101, 1302, 1402-w, and 1602-w, branch metric generator 2004 outputs a set of branch metrics $M_k(S \rightarrow S')$ for transitions between states S and state S' of the decoder. ACS 2005 outputs the ACS results to traceback circuitry 2006, and the transition metrics to starting point 2007. Traceback circuit 2006 outputs the symbols decided by sequence detector 2000, in

reverse chronological order, and LIFO 2008 reverses the order of those symbols to output symbol stream $\hat{a}_{k,w}$.

[0190] The output signals $S_{k,w}$ outputted from ACS 2005 to feedback section 2003 are given by the traced back sequence from traceback circuit 2006.

[0191] From Figure 20A, if the past L symbols have been properly decoded than the influence of intersymbol interference will be completely canceled. In general, feedback section 2003 can be any feedback structure. An example feedback section is feedback section 811 shown in Figure 8. Feedback section 811, however, includes M multipliers (multipliers 806-1 through 806-M) and an M-input adder (adder 807) for the case of M ISI symbol cancellation.

[0192] One embodiment of feedback section 2003 includes feedback sections like feedback section 811. For Gigabit Ethernet, at symbol rates of 125 Mhz on each wire, the timing constraints of the sequence detector are severe. When "per-survivor processing" (*see, e.g.,* sequence detector 1300 of Figure 13) is used, feedback section 2003 must include feedback section 811 repeated for each state of equalizer 2000 because the final state of equalizer 2000 is not determined until the latest decision of ACS 2005 (i.e., the parameters $S_{k,w}$ are determined by ACS 2005).

[0193] Figure 20B shows an embodiment of feedback section 2003 that includes a look-up feedback section 2100. Look-up feedback section 2100 outputs an output signal a''_k from look-up values ξ_1 through ξ_Q in response to the ACS parameters S_{k-1} through S_{k-L}. Look-up values ξ_1 through ξ_Q can be preloaded into feedback section 2100 or may be periodically adaptively chosen and read into feedback section 2100. Values for look-up values ξ_1 through ξ_Q for two ISI symbols and a PAM-5 symbol alphabet are given above in Equation 35.

[0194] The embodiments discussed above are exemplary only and are not intended to be limiting. One skilled in the art will recognize multiple variations of these embodiments that are intended to be included within the scope of this disclosure. As such, the invention is limited only by the following claims.